How to build a brain from scratch

Christopher Summerfield

Department of Experimental Psychology
University of Oxford
Oxford, UK

This advanced option course discusses the search for a general theory of learning and inference in biological brains. It draws upon diverse themes in the fields of psychology, neuroscience, machine learning and artificial intelligence research. We begin by posing broad questions. What are brains for, and what does it mean to ask how they “work"? Then, over a series of lectures, we discuss parallel computational approaches in machine learning/AI and psychology/neuroscience, including reinforcement learning, deep learning, and Bayesian methods. We contrast computational and representational approaches to understanding neuroscience data. We ask whether current approaches in machine learning are feasible and scaleable, and which methods - if any - resemble the computations observed in biological brains. We review how high-level cognitive functions - attention, episodic memory, concept formation, reasoning and executive control - are being instantiated in artificial agents, and how their implementation draws upon what we know about the mammalian brain. Finally, we contemplate the outlook for the future, and whether AI will be “solved” in the near future.

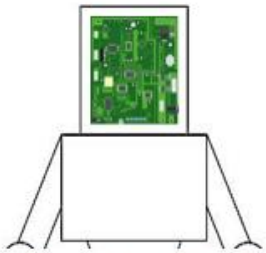
Contents

1. Building and understanding brains

1.1 Introduction

This course is written primarily for students and researchers working in the fields of psychology and neuroscience, who wish to understand more about recent advances in machine learning and artificial intelligence research.

Reverse engineering the brain



Engineering the brain

Psychology: to understand the organisation of behavior and its foundations in cognition

Neuroscience: to understand neural coding and computation, and localise brain function

Artificial Intelligence: to build intelligent information processing systems *in silico*

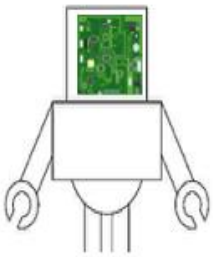
Machine Learning: to use statistical principles to optimize information processing systems

Let me begin with some definitions. Researchers in the field of cognitive and experimental psychology typically aim to understand the organisation of behaviour in humans and other animals, with a particular focus on how cognitive processes give rise to behaviour. Neuroscientists are concerned with the implementation of mental activity in the biological processes of the brain. Their research address questions such as how stimuli or actions are coded in neural circuits, how whole-brain networks contribute to perception, cognition and action. Both psychologists and neuroscientists construct predictive models of brain function and evaluate their models by comparison with empirically observed behavioural and neural data.

By contrast, researchers in the field of artificial intelligence (AI) have the goal of building intelligent artificial systems, for example implemented as an autonomous agent exhibiting complex behaviours in a virtual environment (e.g. an avatar in a video game) or in the real world (e.g. a robot or driverless car). Defining what constitutes “intelligent” behaviour is challenging but some insight can be gained by examining the sorts of problems that are being addressed today (in 2018): researchers are attempting to build systems that can recognise

faces and objects with human-level expertise; that can translate fluently between the world's languages; that can drive a car autonomously on the roads; or that can dextrously manipulate complex objects in an industrial setting. Machine Learning (ML) is the name given to a field that initially grew at the intersection of statistics and computer science, and which uses optimisation techniques to make inferences from big data. Over the last 5-10 years, machine learning has provided some of the most promising solutions to complex problems faced by autonomous agents and is thus rapidly merging with the field of AI. The terms AI/ML will largely be used interchangeably here.

One might think of the goal of AI/ML researchers as "engineering the brain" in contrast to that of researchers in psychology and neuroscience, whose work is aimed at "reverse engineering the brain"[1].



August 31st 1955

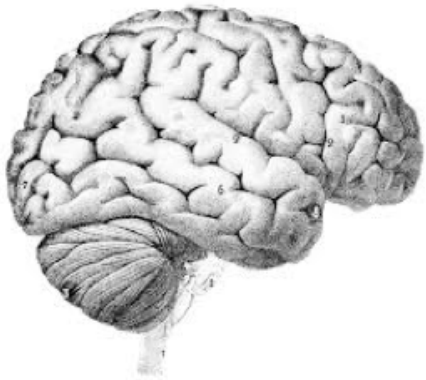
"We propose that a 2-month, 10-man study of artificial intelligence be carried out in the summer of 1956 at Dartmouth College…The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it"

Let us continue with a bit of history. The field of AI may be said to have been born in the mid-1950s, with a summer conference that took place in Dartmouth, MA in 1956[2]. Delegates of this conference included many of the foundational figures in early AI research, including Marvin Minsky, Alan Newell, and Herb Simon, as well as key figures in the development of 20th century mathematics such as Ray Solomonoff (inventory of algorithmic probability) and Claude Shannon (father of information theory). The initial proposal was that over the course of 2 months, 10 key figures would assemble to contemplate how to build a machine that could

[1] Helpful introductory reviews concerning the relationship between Psychology/Neuroscience and ML/AI are include Cox, D.D., and Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. Curr Biol *24*, R921-R929, Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-Inspired Artificial Intelligence. Neuron *95*, 245-258, Marblestone, A.H., Wayne, G., and Kording, K.P. (2016). Toward an Integration of Deep Learning and Neuroscience. Front Comput Neurosci *10*, 94.

[2] An interesting book that mixes science and gossip associated with this period in history is Anderson, J.A., and Rosenfeld, E., eds. (2000). Talking nets: An oral history of neural networks (MIT Press).

behave autonomously. The foundational premise was that all aspects of biological intelligence can, in principle, be distilled into an artificial system. The purpose of the conference was to decide how to build such a system. It was proposed that "substantial progress can be made…within a single summer".



Human brain has ~10^{11} neurons each making an average of 10^3 synapses

Runs on only ~20 watts (20% of total bodily energy consumption)

Commonly stated[1]: "human brain is the most complex device in the known universe"

Engineering the brain is a challenge

[1]Christoph Koch, head of the Allen Institue for Brain Science

However, it turned out that rebuilding biological intelligence from scratch – and in particular, human intelligence – was a project that would require more than ten clever people and a single summer. More than 50 years (and numerous setbacks) later, we are still arguably a long way from recreating in an artificial system even the level of intelligence displayed by (say) a mouse. In hindsight, perhaps that is not so surprising. Brains are complex organs. The average human brain, for example, contains ~10^{11} neurons each of which has synaptic connections with an average of 10^3 other cells. Christoph Koch, the renowned neuroscientist and head of the Allen Institute for Brain Science has suggested that the human brain is perhaps the "most complex device in the known universe" (and certainly the "most complex known device in the universe") Yet more surprisingly, this fantastically complex organ runs on only ~20 watts of electricity, i.e. an energy budget comparable to that of a regular lightbulb.

Paris is the capital of France

2 x 3 = 6

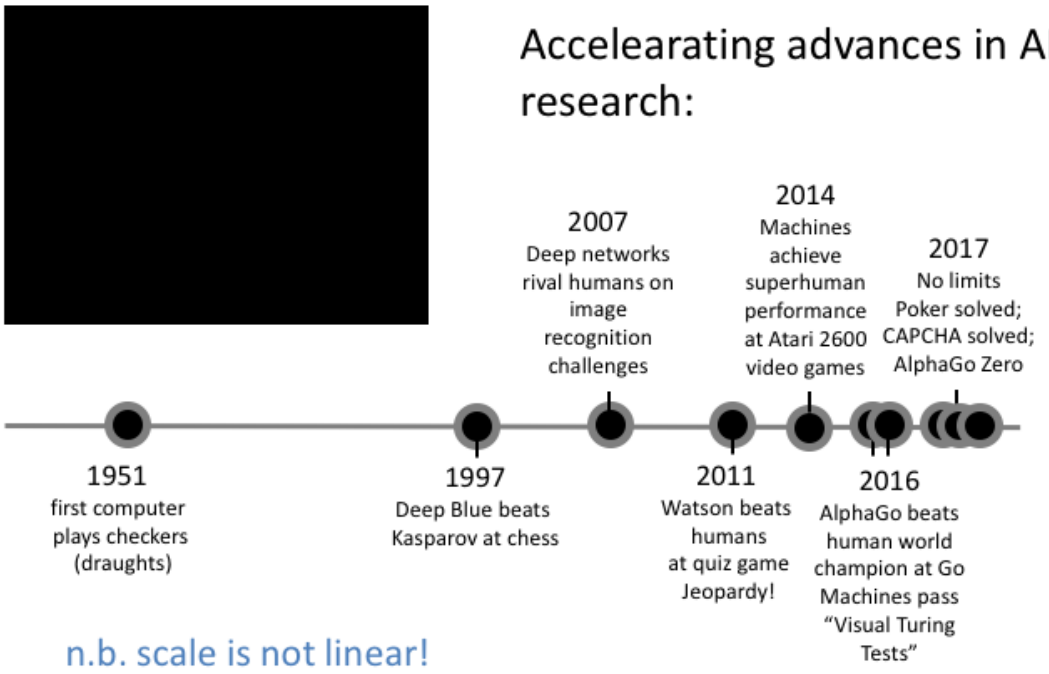
Swans are birds; all birds lay eggs…*ergo* swans lay eggs

Humans know stuff.

The challenge is to build artificial system that acquires and uses a rich body of conceptual knowledge

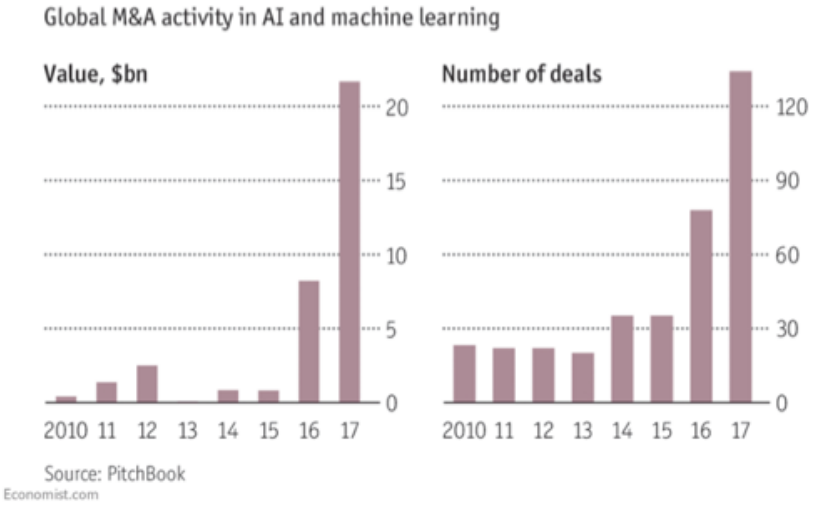
Another way of envisaging the scale of this challenge is to consider the richness of the human mind. Humans are not just able to perform complex computations, like a calculator. Rather, the distinctive property of human cognition seems is the rich conceptual knowledge that underlies our behaviour. We know that you get wet when it rains unless you are carrying an umbrella, and that you can't make a salad out of a pair of socks. We can also use that knowledge to make new inductive inferences. For example, if I tell you that swans are birds and that all birds lay eggs, you can infer that swans lay eggs. How are we ever going to understand how to build a system that can learn, tabula rasa, to encode such rich knowledge of its environment, as humans do?

1.2 Recent advances in AI research



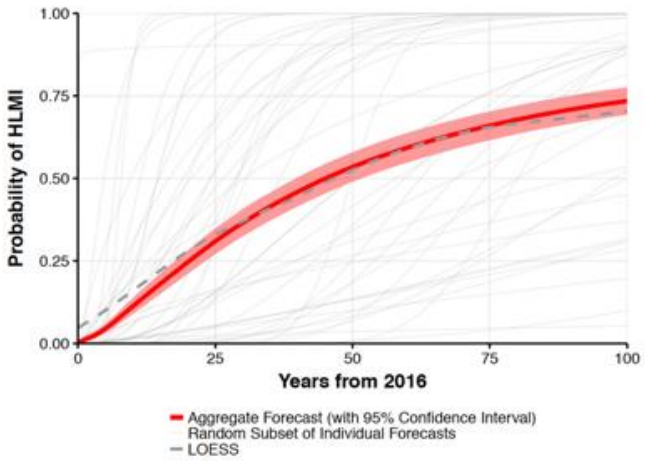
It's 2018. Unless you have been living under a stone, you probably haven't failed to notice that there is a lot of interest in AI research at the moment. Across this course, we shall explain why there is such interest, and discuss what promise these new advances hold for the future. However, for the current purposes let us just briefly examine some recent successes in the field of AI research. In the figure above, the (nonlinear) timeline shows some landmark advances in AI systems. The salient feature is that whilst some major goals were met prior to the year 2000 (for example, IBM's Deep Blue becoming world chess champion by beating Gary Kasparov), the majority of recent advances cluster at the right of the timeline. The last 5 years have seen artificial systems beat humans at the quiz game Jeopardy!, achieve superhuman performance at Atari 2600 video games by learning from pixel inputs and the game score alone (see video), master the ancient board game of Go, previously thought to be at least 10 years out of reach of current AI systems, beat humans at No limits hold 'em poker, and so on[3]. Things are moving fast. This is why there is lots of excitement at the moment, and discussions about whether AI will be "solved" in the near(ish) future.

[3] See the following publications Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., *et al.* (2015). Human-level control through deep reinforcement learning. Nature *518*, 529-533, Moravcik, M., Schmid, M., Burch, N., Lisy, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., and Bowling, M. (2017). DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. Science *356*, 508-513, Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.* (2016). Mastering the game of Go with deep neural networks and tree search. Nature *529*, 484-489.



Industry is pouring money into AI startups

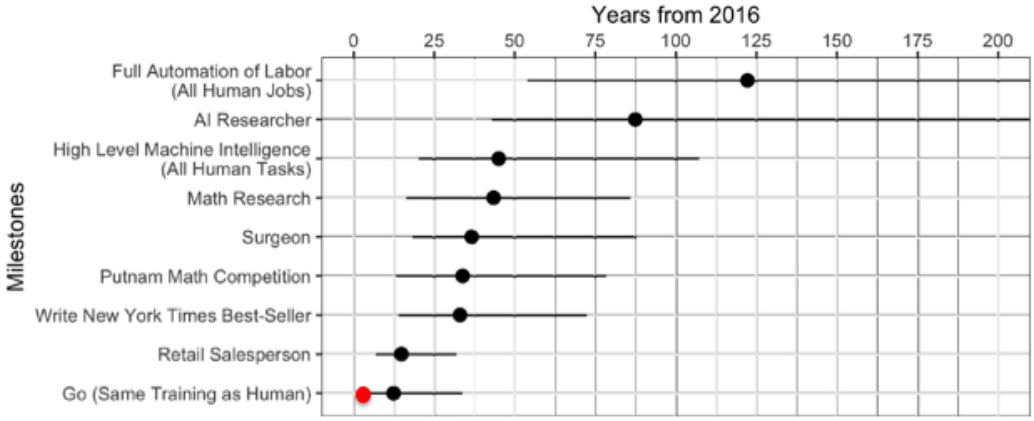
In passing, I note that there is also a lot of money changing hands. These graphs show numbers (right) and value (left) of mergers/acquisitions in the fields of AI and machine learning. The largest bar is for 2017. The bar for 2018 will be even larger. AI is currently a lucrative field to be working in.



Even money that human-level AI will be achieved ~50 years (i.e. in your lifetime)

Despite this enthusiasm, you might think that the goal of "solving" AI – that is, of building an artificial system that exhibits the same level of intelligence as an average human – is still impossibly far off. Of course, you could be right – but experts disagree. The figure above shows

the results of a survey[4] conducted at the 2016 Neural Information Processing Systems (NIPS) conference, an annual meeting of ~5000 machine learning researchers. The delegates were asked to rate the probability that human-level machine intelligence (HLMI) would be achieved at various points in the future. The average forecast estimated that there was a 50% chance that AI would be "solved" by 2066, that is, 50 years from the data of the survey. That is probably within the lifetime of many people reading this. Given the extraordinary societal changes that are likely to ensue from the development of HLMI, this is a quite astonishing prediction.



Even this may be an underestimate!

You might reasonably think that AI researchers have a vested interest in responding optimistically to such a survey – after all, their job depends on it. But there is one data point that has subsequently come to light that suggests that the researchers' predictions might in fact be overly *pessimistic*. Among the more detailed questions on the survey, participants were asked to forecast the likely future date of specific advances, such as when an AI system will be able to write a New York Times bestselling novel (2046, apparently). Recall that in 2016, news had just broken that AlphaGo, an AI system that played Go built by Google Deepmind, had beaten former European champion Fan Hui in a 5-game match, but only after being trained on moves taken from a corpus of >30 million human games. One question thus was: "when will an AI system be able to master Go after having experienced only the same level of training as a human?". The average prediction was that it would take ~12 years. In fact, that challenge was met within a year, with the publication of a paper describing "AlphaZero", a related network that mastered Go, Chess and Shogi (Japanese Chess) after learning entirely by playing itself over the course of 3 days[5], and even surpassed the performance of the original AlphaGo that was trained with human data.

It is these recent advances, and others not detailed here, that underpin the excitement currently surrounding AI in the media. In many countries, this excitement is translating into public policy initiatives. For example, in 2017 the Chinese government announced funding to

[4] https://arxiv.org/pdf/1705.08807.pdf

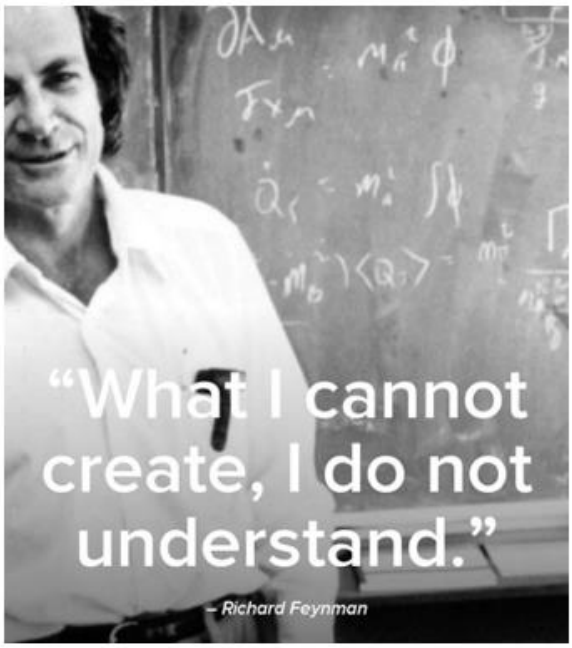
[5] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A.*, et al.* (2017). Mastering the game of Go without human knowledge. Nature *550*, 354-359.

create 100,000 new PhD places for AI research in their national universities. Yes, 100,000. That's not a typo.

1.3. Biological and artificial brains

Biologically-inspired technology has a chequered history. Sometimes, to solve a problem it's best to dispense with solutions from biology.

In this course, I am going to advance an argument about the future of AI. It is an argument that has historically been quite unfashionable, but, I think, is coming back into vogue. That argument is that in order to build artificial brains, we can look to biology for inspiration. The reason that this argument has failed to gain traction is that there are important counterexamples in other domains, where attempts to copy biology failed to yield new workable technologies. The most oft-cited example is that of human flight. For many years, humans attempted to build flying devices that drew inspiration directly from the wings of birds. These were not successful. Indeed, it was only by completely ignoring biology and pursuing another path (i.e. the invention of the jet engine) that human flight became a ubiquitous feature of our modern lives. Many AI researchers still feel that the same principle holds for building brains. Biological brains are unpredictable, error-prone and tend to learn suboptimal policies (choosing the wrong insurance policy, and marrying the wrong person); why not dispense with these and allow principles of statistical optimisation to do a better job? However, as we shall see throughout the course, there are numerous examples where major progress in AI was achieved precisely because researchers directly imported an idea gleaned from the study of animal behaviour and/or biological brains.



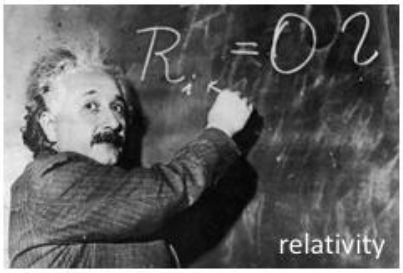
Principle:

to really understand something, I have to be able to build it

Richard Feynman, nobel laureate in physics

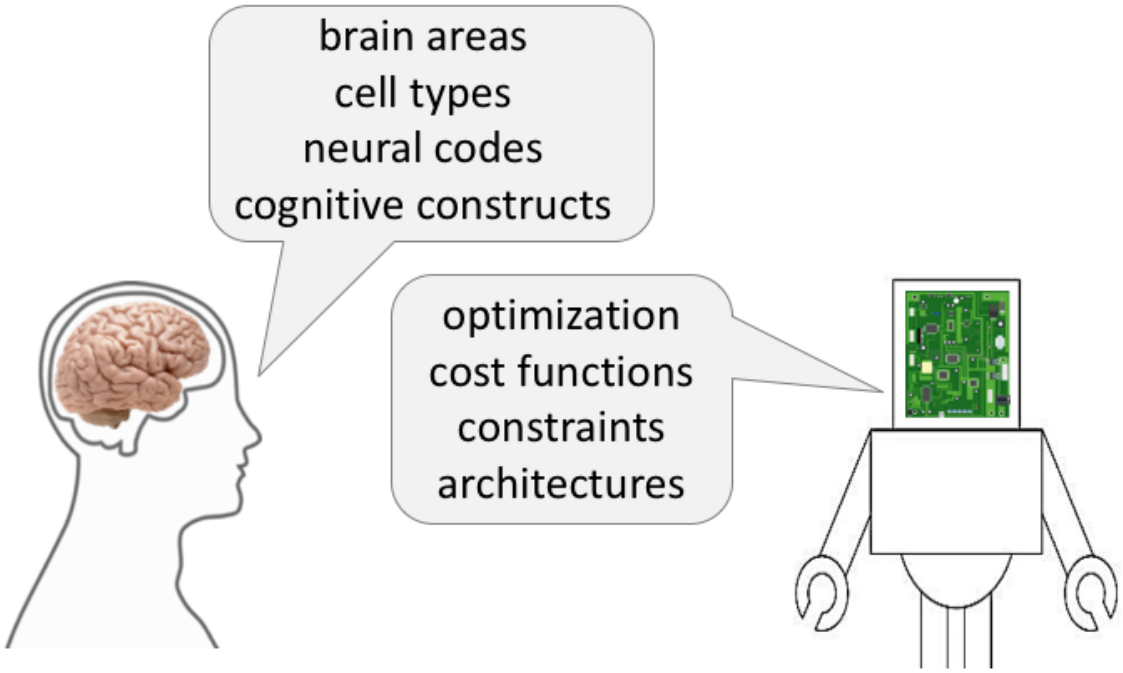
However, the major aim of this course is not to describe how biology can inform AI, but vice versa – to emphasise what psychologists and neuroscientists can learn from recent progress in ML/AI. The principle that best summarises this view comes from a quote by Nobel Laureate Richard Feynman, who famously said "what I cannot create, I do not understand". In other words, in order to *truly* understand something, you have to be able to build it. Anyone who has attempted to simulate a brain process or behaviour using a computational model knows that this is true. Here, we are going to argue that psychologists and neuroscientists have a lot to learn from AI researchers who are trying to build brains, because doing so forces them to really understand how a brain works.

Psychology and neuroscience currently lack a **general theory of brain function**. Paradoxically, this is being developed in AI, not in our field!



Past attempts (Behaviourism, Connectionism, Bayesianism) have all been inadequate, or offered limited explanatory power

In fact, one might make a case for an even more outlandish claim. Over the course of the 20th century, some disciplines developed *general theories* that provide a framework for all new understanding in that field. For example, biologists all subscribe to the theory of natural selection, and physicists (more or less) buy into Einstein's standard model of relativity. However, such a general theory has been elusive in psychology and neuroscience. We conspicuously lack a *general theory of brain function*. There have been various past attempts to formulate such a theory – one might point to the past movements of Behaviourism, Connectionism, or Bayesianism – but these have all (I would argue) been shown to have been inadequate as full descriptions of brain function. Currently, there is virtually no attempt being made to formulate such a general theory in the fields of psychology and neuroscience. Strikingly, however, such a general theory is being formulated, discussed and implemented as we speak – but not in the corridors of psychology and neuroscience departments. Rather, it is AI researchers that are beginning to build such a theory. One of the major aims of this course is to communicate that theory back to the social and biological sciences.

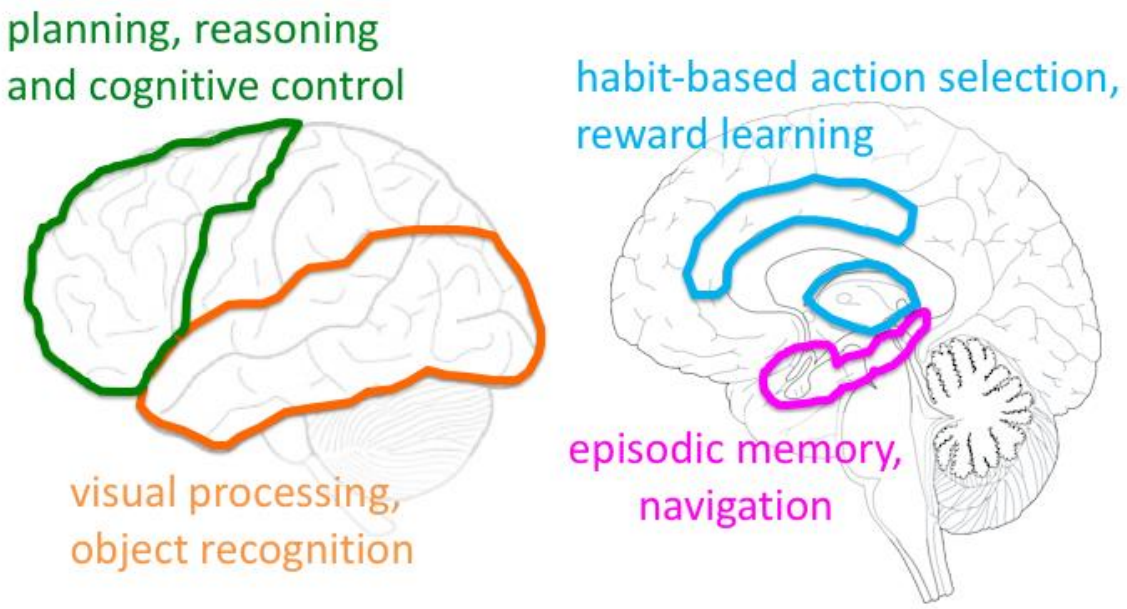


Marblestone et al 2016

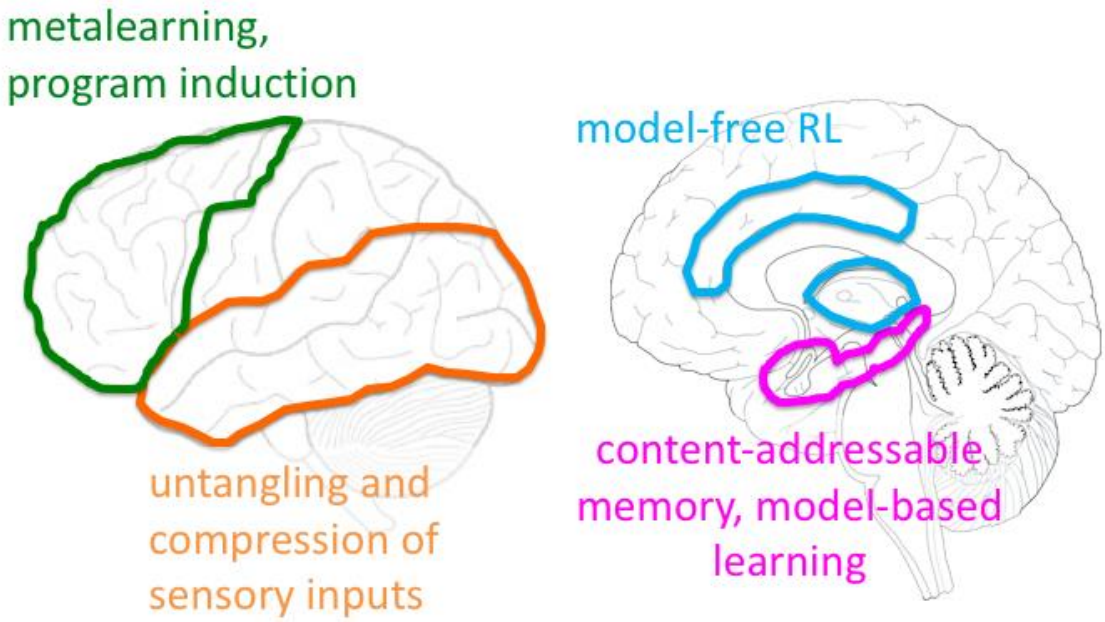
Why do psychologists/neuroscientists and AI/ML researchers not get together more and share their ideas? In part, they do. The fields have gone through rich periods of cross-fertilisation, dating back to the 1950s. One of the more recent of these periods occurred about 10 years ago, when neuroscientists awoke to the rich theoretical framework provided by reinforcement learning, and its potential as a theory of brain function (notably, this theory was in turn inspired by much early work aimed at understanding animal learning, conducted in psychology departments). However, one salient barrier to knowledge exchange is that researchers in psychology/neuroscience and AI/ML speak different languages[6]. The language used to describe biological brains typically alludes to the constituent parts of a neural system, that is cell types and brain areas; and the coding principles and cognitive constructs that allow the system to function. By contrast, researchers in ML/AI have converged on a different way of thinking about neural systems, which is guided by principles of function optimisation. A brain is a model of function which is optimised to minimise some cost, under a set of constraints implemented in processing architectures. These terms have yet to become part of the lingua franca of psychology/neuroscience research.

In part, one might argue that the gap arises because researchers in machine learning are principally concerned with – the clue is in the name – learning. Our techniques for measuring learning are grossly impoverished in psychology and neuroscience. This is largely due to technical limitations. Learning takes time. It is logistically complex to run studies in which participants return day after day to undertake a task. In experimental animals, where repeated experimentation is possible, there are other challenges. For example, single-cell recording techniques do not permit the same cell to be tracked over more than a day or two. Of note, this may be changing with the advent of new optical imaging methods.

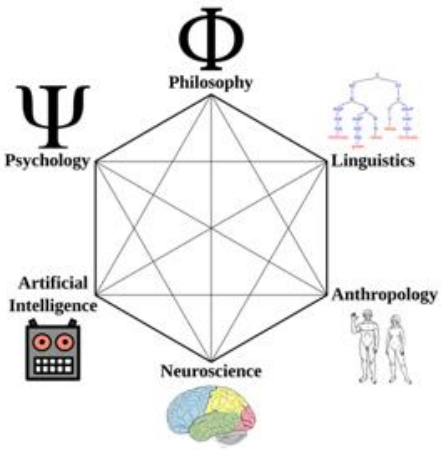
[6] Marblestone, A.H., Wayne, G., and Kording, K.P. (2016). Toward an Integration of Deep Learning and Neuroscience. Front Comput Neurosci *10*, 94.

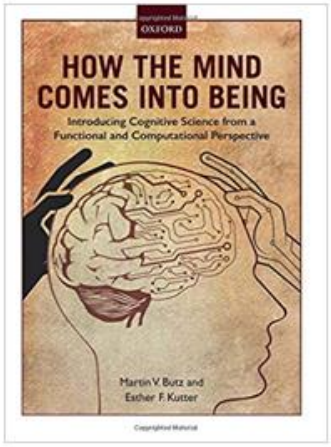


It is possible to illustrate by considering some of the widely accepted theories about the functional neuroanatomy of the primate brain. In the figure above, I illustrate some brain regions and their associated proposed functions. For example, most first year undergraduate students in psychology/neuroscience are taught that the hippocampus and surrounding medial temporal lobe are critical for the functions of episodic memory and spatial navigation, whereas habit-based learning and action selection depend critically on the basal ganglia and overlying medial prefrontal cortex. These theories are backed up by extensive evidence from lesion studies, single-cell recording techniques, and functional brain imaging studies (as well as a plethora of new methods from synthetic biology).



However, if we look at the same functions from the perspective of AI/ML, they have very different names. There is general agreement that episodic memory and navigation require "one-shot" learning and/or a "model of the world", whereas habit-based action selection is known as "model-free RL". To communicate better, researchers in the two fields need to establish a common language.



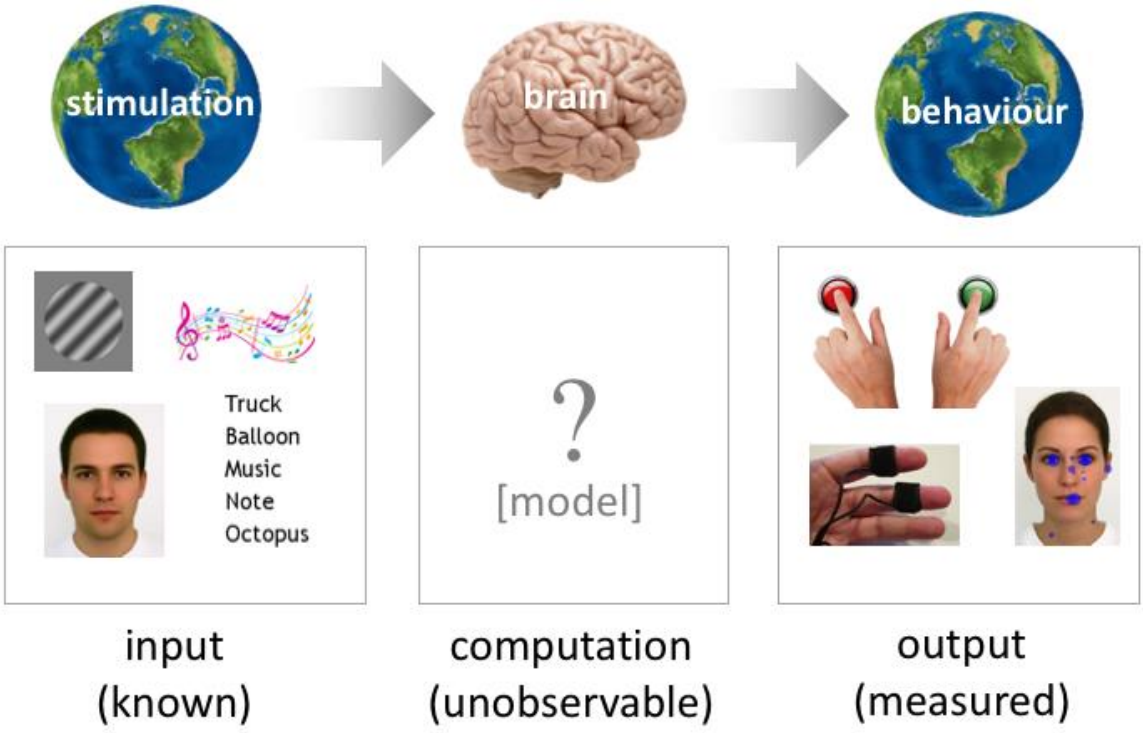


Cognitive Science is a multidisciplinary endeavor that seeks to both understand and build the brain, viewing brains as information processing systems

Butz & Kutter 2017

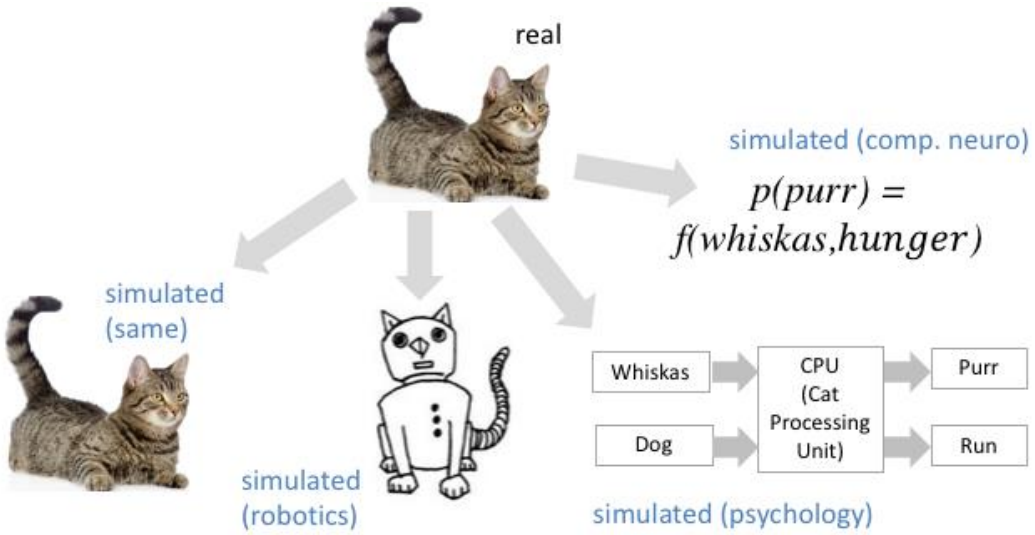
However, before we assume that researchers in psychology/neuroscience and AI/ML never share ideas, let us not forget that there is a discipline where they can (and do) come together. It's known as cognitive science. Cognitive science was the real fruit of that Dartmouth conference in the 1960s and has since become a rich field that synthesises ideas not just from psychology, neuroscience and AI, but also from anthropology, linguistics and philosophy. A recent textbook[7] offers a contemporary perspective on cognitive science and may be a useful accompaniment to this course.

1.4. The computational approach



Next, let us consider the general framework under which psychologists and neuroscientists conduct their research. The bread and butter of an experiment in our field are the independent variables we manipulate, and the dependent variables we measure. As psychologists, we typically control the stimuli in an experiment. For example, we might present participants with a psychophysical stimuls (e.g. a grating), an image (e.g. face), a set of items to be remembered (e.g. a list of words) or an auditory stimulus (such as a piece of music to be rated). We collect behavioural data, whether in the form of choices or reaction times, eye movements, or physiological measures such as galvanic skin response. The goal of our research is to understand some computational principle by which inputs (stimuli) are converted into outputs (responses). That computational principle is unobservable and must be inferred from the relation between inputs and outputs. We call the resulting principle a model.

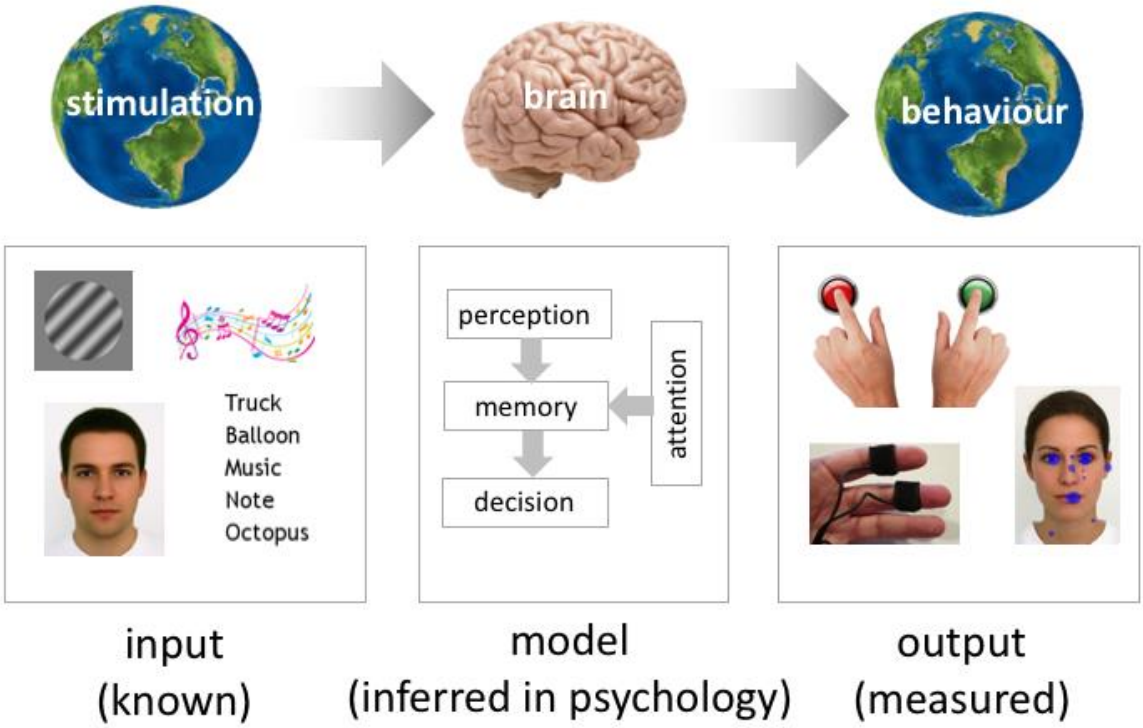
[7] Butz, M., and Kutter, E.F. (2017). How the mind comes into being: Introducing Cognitive Science from a functional and computational perspective (Oxford University Press).



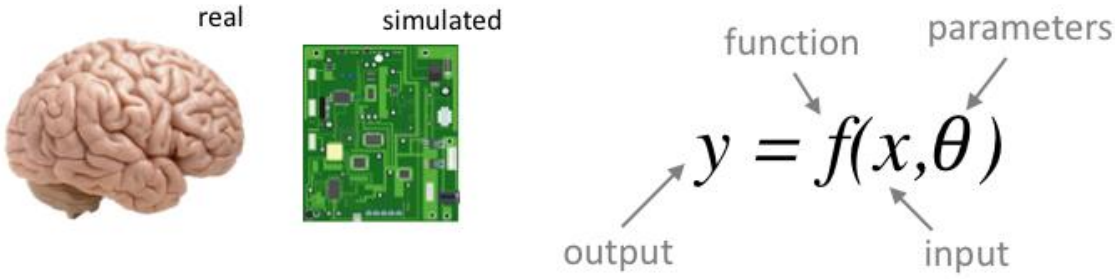
"The best model of a cat is another cat, or preferably the same cat"

Norbert Weiner
Founder of Cybernetics

What do we mean by a "model"? A model is a simulation of the computational processes that intervene between inputs and outputs. The success of our model is evaluated by how well it fits the data. An ideal model will perfectly recreate the conversion of stimuli into responses and allow us to make highly accurate new predictions about how people will behave. To quote Norbert Weiner, the founder of the field of cybernetics: "The best model of a cat is another cat, or preferably the same cat". If our model is literally the same as the actual processes we are trying to understand, the we have succeeded in understanding the brain. Of course, given the abovementioned complexity of brains, this is unlikely. It's more likely that our model will be an approximation. In different subdisciplines within psychology and neuroscience, different approaches to this approximation are made. For example, cognitive psychologists have tended to favour the sort of model shown above, in which component processes are given descriptive labels that eschew precise quantification of how information is converted (or transduced) from an input to an output. In psychophysics and sensory neuroscience (as well as machine learning), the favoured model often has a compact mathematical description, i.e. it can be written as a set of closed form equations. This offers the opportunity to make clearer predictions about how humans or experimental animals will behave in novel settings. Unfortunately, however, given the complexity of the mind and the richness of behaviour, computational neuroscientists have largely limited their ambition to understanding relatively simple processes, such as early sensory transduction or skilled motor control, avoiding any detailed modelling of central or executive processes that most consider integral to intelligence.



So, returning to our schematic, we can include the sorts of models built by cognitive psychologists.

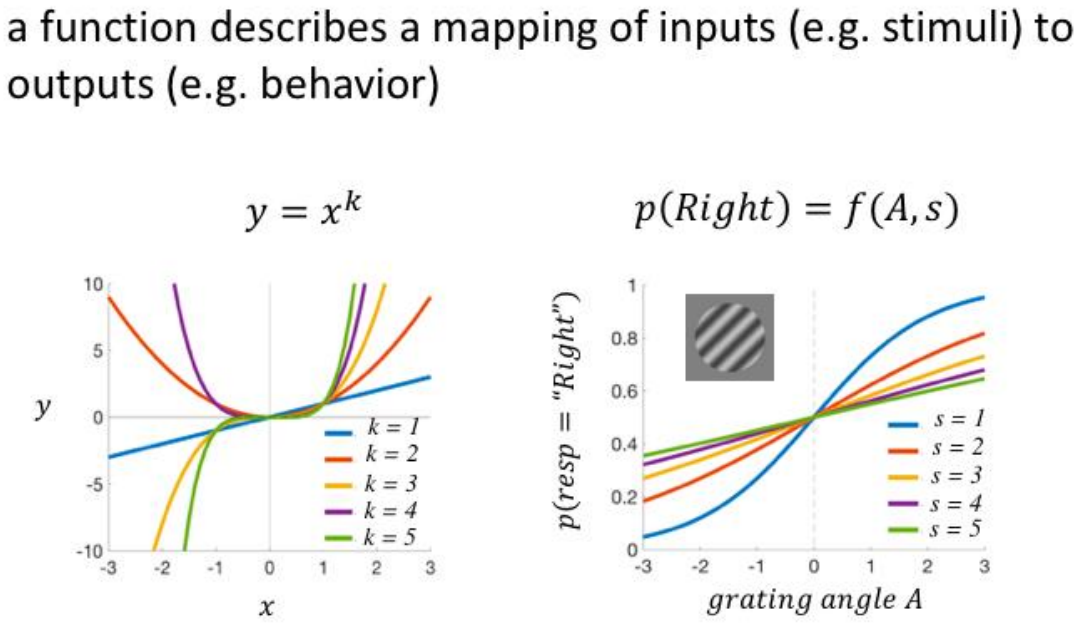


Mathematics provides a compact and precise language for describing input-output relations

Computational models are just parameterized functions (or programs, expressed in computer code) that transform inputs into outputs

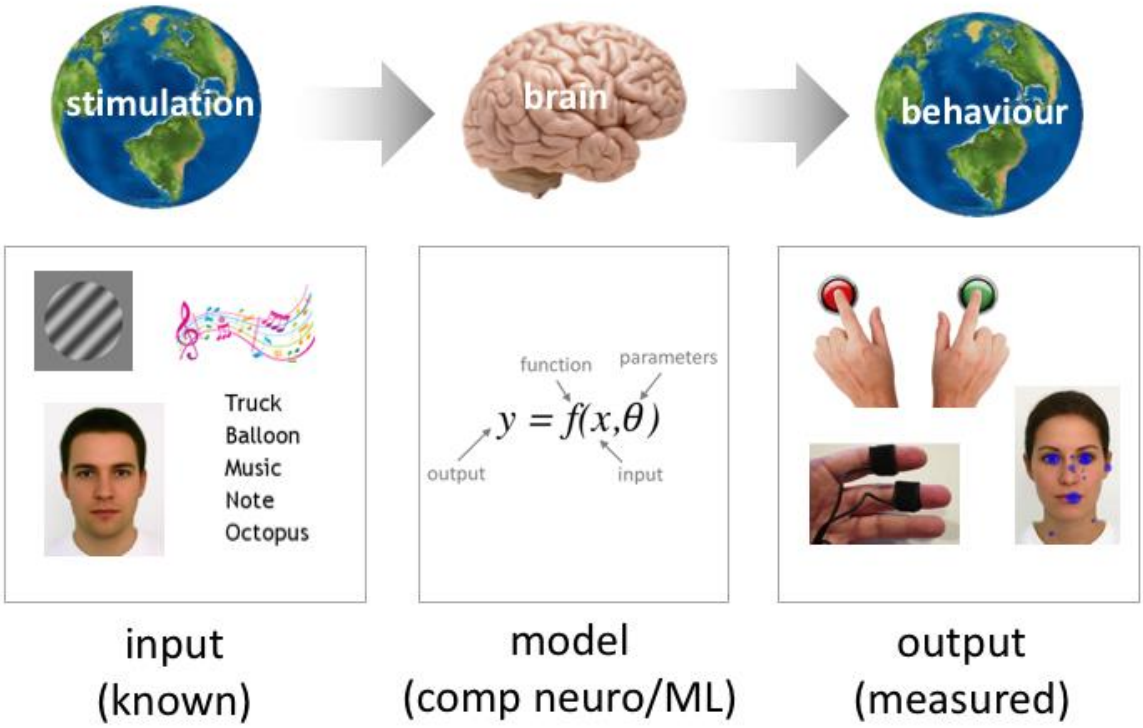
However, as alluded to above, mathematics can provide a more precise language for formulating a theory of brain function. A useful approach is to characterise the information

processing steps that intervene between sensation and action as a function, that takes as inputs some sensory data x and processes it using a model characterised by parameters θ to provide a behavioural output variable y. Our model now takes the form of a piece of computer code, that transforms simulated inputs into simulated outputs according to some principles that we think embody the workings of biological brains.

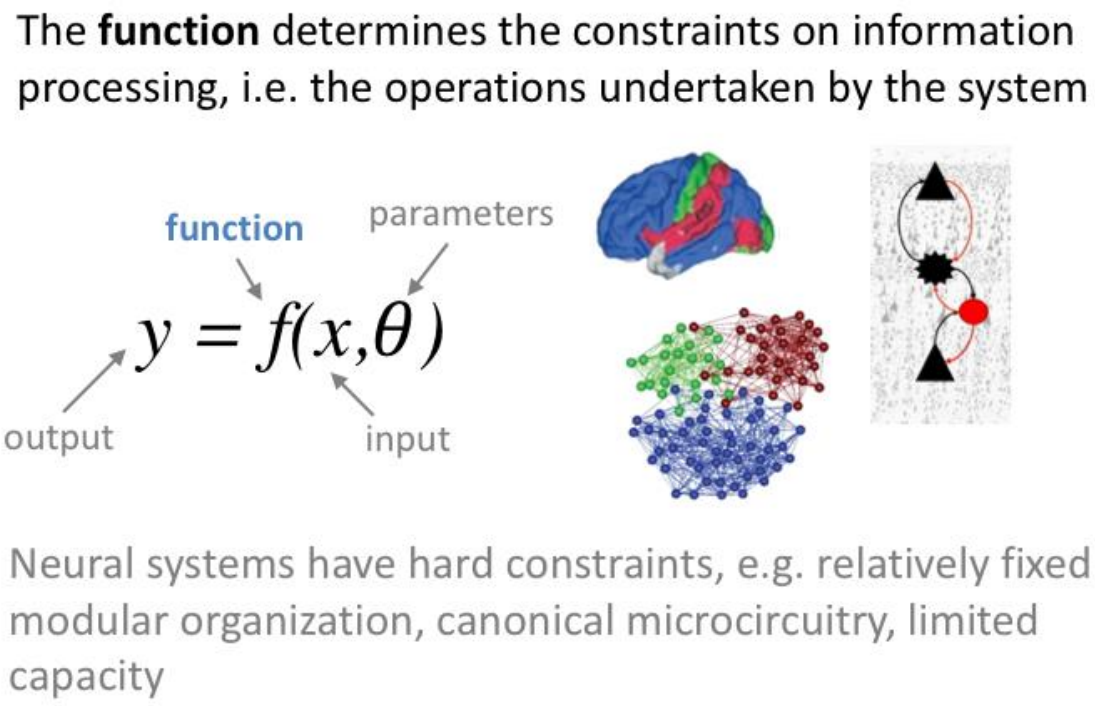


Before proceeding further, let us just unpack what is meant here by a "function". An example function is shown on the left. This function performs an exponentiation of the input data x, producing an output y that is simply x^k where k is the exponent. Here, k is a free parameter whose value determines the precise mapping from x to y.

On the right, I show another example of a function, but this is closer to the sort of model that might be used in psychology. Now, the output y corresponds to a probability of a given response, i.e. the fraction of times that a participant responds that a grating is tilted "right" (or clockwise) of the vertical meridian. However, now the function that transforms the input x (e.g. the tilt of the grating) into output y is a logistic (sigmoidal) function with slope s. In fact, the formula for this function is $y = \frac{1}{1+e^{-sx}}$, describing the canonical for the psychometric function, but this is not crucial for our purposes. The point is that by positing a (sigmoidal) function that maps inputs onto outputs we are making a claim about the information processing steps by which a psychophysical observer makes judgments of orientation.

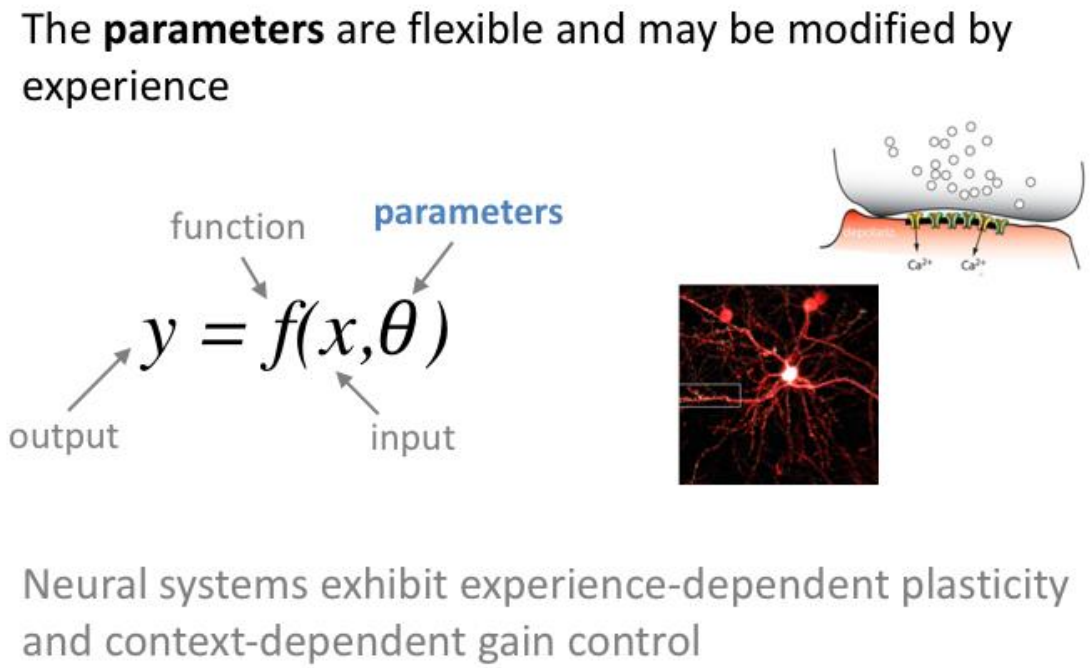


So, for completeness, we add to our schematic the sort of model that is typical in computational neuroscience (and psychophysics).



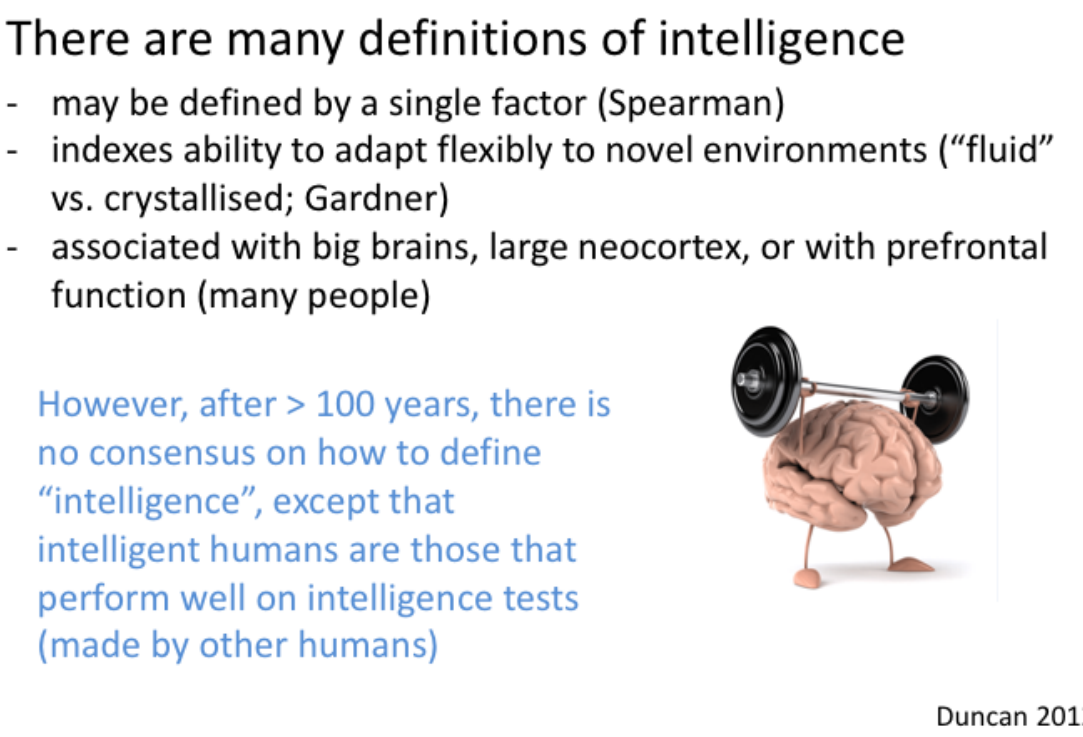
This way of thinking of the brain might seem rather alien to those who are unaccustomed to it. What do we mean that the brain can be described as function with free parameters? What does this sort of description have to do with neurons and synapses? Well, as we shall see during this course, the function describes the neural architecture by which (sensory) information is

transduced *en route* to a response. In the language of machine learning, it provides the *constraints* that delimit the behaviour of the system. Information processing in biological brains is structured in stereoptyed ways, and it is the job of the computational neuroscientist to discern the nature of this structure. We know that neural systems have strong constraints, and that these tend to be shared across individuals virtually irrespective of their life experience. For example, nobody has unlimited working memory capacity, and every healthy individual has a hippocampus. How these constraints arise is visible in a close examination of the brain's architecture – its neuroanatomy. For example, the canonical microcircuit is a repeating motif across 6-layer cortex in mammals. Vertebrate brains also have a highly stereotyped modular organisation, with different regions (basal ganglia, hippocampus, neocortex, processing information in distinct ways and thus providing differing constraints. Neural systems are not undifferentiated processing machines, but have structured processing steps, each of which transform information in different ways.



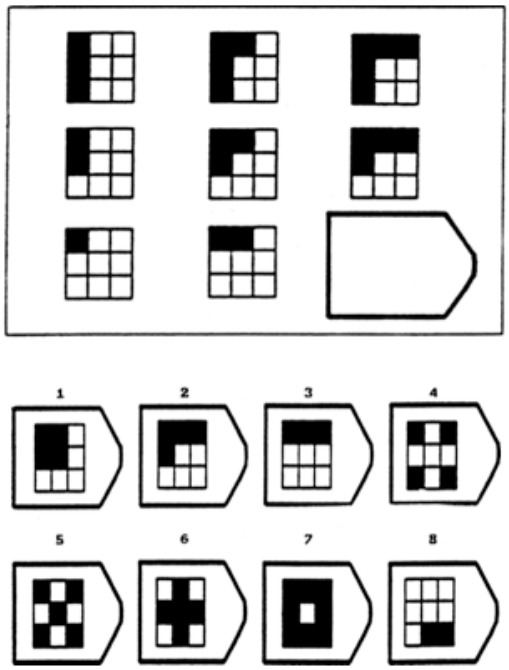
So what about the parameters? The parameters give a processing architecture its flexibility. Given a specific function (or architecture), the way that inputs are converted into outputs will differ according to its parameters (compare the coloured lines in the figure about functions above). In neural terms, we can think of the parameters as the internal settings that are unique to an individual, such as the strengths of individual neural connections (synaptic weights) or the configuration of the neuromodulatory gain control systems. In other words, the parameters of the information processing system define its *content*. These are modifiable by experience (unlike, in general, the architecture), for example because synapses are plastic and change during learning. For example, the connections in your cortex and hippocampus may be different from mine, because you have had different past experiences to me, and so faced with the same inputs, you might behave differently. Another way of putting it: your "knowledge" is the parameter setting in your brain.

1.5. Definitions of intelligence



The nature of human intelligence has been a prominent theme in psychology for nearly 100 years (animal intelligence is also hotly debated, but typically in the sister field of behavioural ecology)[8]. The measurement of intelligence began in the early 20th century, when the US government sought a way to select among prospective immigrants. Major themes in the study of intelligence emerged over the course of that century, including Spearman's notion that intelligence may be a unitary construct, that can be measured with a single parameter, known as g. Evidence for this view came from the finding that performance across a range of cognitive tasks tends to be highly correlated, i.e. those that perform well on attention tasks also perform well on memory tasks. Of course, it is unclear whether this is because there is really only "one intelligence" or it follows from the fact that our tasks tap into a common mixture of factors, i.e. there are actually several uncorrelated variables that determine intelligence, but attention and memory tasks jointly index several of them at once. Subsequently Cattell introduced the notion of "fluid" and "crystallised" intelligence, the latter roughly pertaining to the ability to reason logically in novel settings, and the latter determined by the body of knowledge acquired by experience (think: the power of the function and the setting of the parameters, respectively). Later, Gardner expanded this definition to consider "multiple intelligences" (he dreamt up eight). In the later 20th century, the development of brain imaging methods led to the suggestion that intelligence is associated with executive function and the prefrontal cortex. Largely, however, intelligence research has been an exercise in armchair theorising with only weak empirical evidence to support the various claims, and it is probably fair to say that after 100 years we still lack a clear definition of what might constitute "intelligence" for humans, other animals, or for machines. I would argue that this is likely to pose a major roadblock for AI research. How do we know if we have built an AI, if we don't know what intelligence is?

[8] A nice summary in this book: Duncan, J. (2012). How intelligence happens (Yale University Press). For work on animal intelligence, I recommend this book: de Waal, F. (2016). Are we smart enough to know how smart animals are? (W. W. Norton & Company).

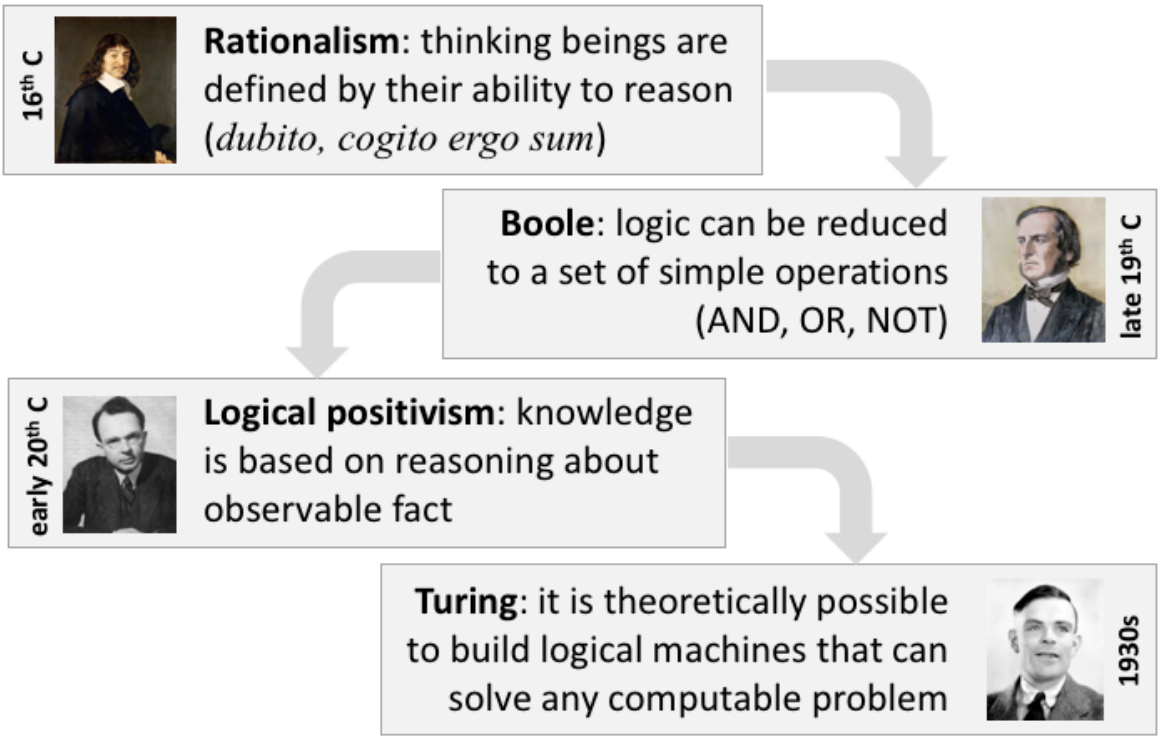


Example from standard intelligence tests:

Raven's progressive matrices tests **abstract nonverbal reasoning**

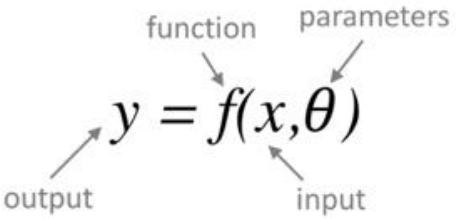
Raven, 1936

Despite this theoretical vacuum, measuring intelligence is an important application of psychology research. Intelligence tests tend to incorporate our (perhaps unexamined) notion of what constitutes fluid intelligence in the 20th century tradition, i.e. it corresponds broadly to the ability to reason logically about novel stimuli. One example is Raven's Progressive Matrices test, in which a series of patterns are shown that follow a progression according to one or more unstated variables. The goal is to identify the missing pattern, as in the example above.



This notion that reasoning ability = intelligence has its foundations well before the maturity of psychology as a discipline. It can be traced back to the rationalist tradition epitomised by Descartes: I doubt, and thus I think, and therefore I am. According to this tradition, reason is the essence of what defines a human. Throughout the late 19th and early 20th centuries, important currents of thought sought to use reasoning as an ontological framework for understanding the world. This began with Boole's seminal work showing that all logic can be reduced to a set of primitive operations (e.g. AND, OR and NOT) and continued with the logical positivist movement, which attempted to find a rational basis for all mathematics using tools from logic. In the mid-20th century, early work in computer science built on this tradition, for example with Turing's 1936 proof that a machine that implements primitive logic (a Turing machine) can in theory solve any computable problem[9].

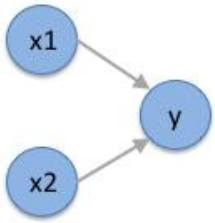
[9] Anyone who is really interesting in what Turing's major contribution to computer science was, including really understanding what a Turing Machine is and how it works, is recommended this book: Bernhardt, C. (2017). Turing's Vision: The Birth of Computer Science (MIT Press).



How can a function like this reason logically?

Example programme implementing AND gate

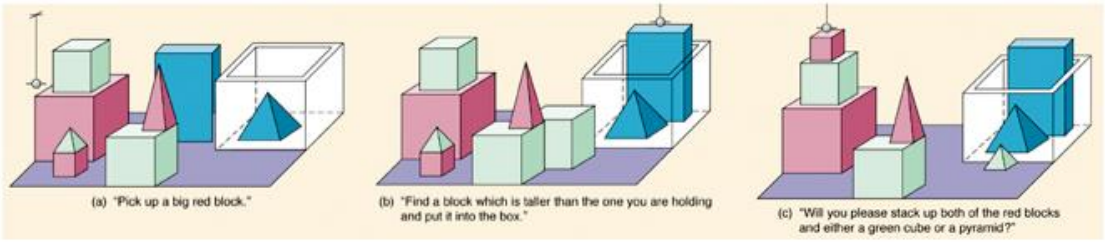
```
1 function y = myANDgate(x1,x2,thresh)
2
3 if x1 > thresh &  x2 > thresh
4   y = 1
5 else
6   y = 0
7 end
```



This programme be described in graphical (or network) form

Brains are large lumps of salt, water and protein, whose cells produce action potentials and whose computational is determined by the connections among neurons. How can a brain implement a logical operation? Recall that our computational definition of how a brain works can be described by a function. As we discussed, a function is a computer programme that converts inputs into outputs according to a set of rules (constraints). Consider a simple network of neurons wired up as in the slide above (blue). The code on the right defines a function that implements an AND gate: if both neurons x_1 AND x_2 are active above threshold, then neuron y will become active.

1.6. Good Old-Fashioned AI (GOFAI)



Early AI systems involved manipulation of symbols through logical rules

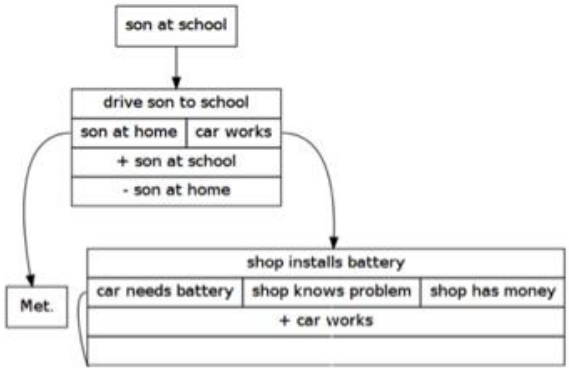
e.g. SHRDLU was able to interactively manipulate objects in a blockworld, and respond to queries

Terry Winograd (PhD supervisor of Larry Page)

Taken together, these intuitions motivated the earliest research into artificial intelligence. It was taken as given that intelligence was defined by the power to reason logically about symbols, ideally about novel stimuli or situations. So the first AI systems involved the manipulation of symbols through logical rules. For example, Terry Winograd (the PhD supervisor of Google's founder, Larry Page) build a system known as SHRDLU that took as inputs the position, shape and size of a series of blocks on a table-top world, and a query or instruction that defined an interaction with this world (e.g. "pick up the block that is next to the blue block"). The system understood a small (~50) but fixed vocabulary of instructions and was able to combine them in novel ways to follow instructions, through a dynamic interaction with the user that resembled a (limited) conversation.

I want to take my son to nursery school. What's the difference between what I have and what I want? One of distance. What changes distance? My automobile. My automobile won't work. What is needed to make it work? A new battery. What has new batteries? An auto repair shop. I want the repair shop to put in a new battery; but the shop doesn't know I need one. What is the difficulty? One of communication. What allows communication? A telephone... and so on....

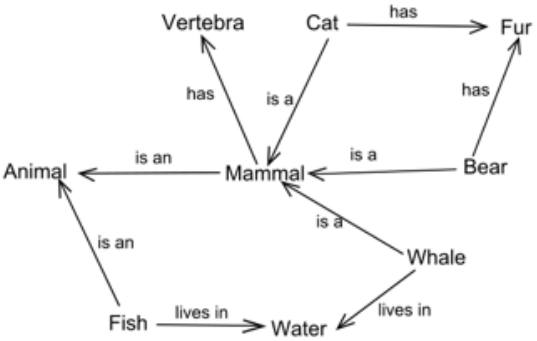
General problem solver follows a logical progression through a means-end search space

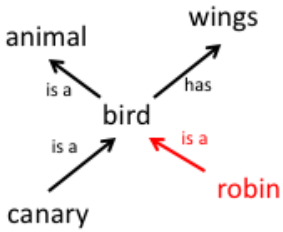


Newell & Simon, 1972

Other, comparable systems similarly engaged in means-end reasoning using natural language as inputs. For example, Alan Newell and Herb Simon – two attendees of the original Dartmouth conference – built the "general problem solver", which was able to reason about any (sufficiently small) set of contentions that could be formulated unambiguously as a directed graph. For example, it could solve relatively restricted search problems such as the Tower of Hanoi. The example above is illustrative of the sort of approach that the GPS took, although in practice it was unable to solve any real-world problems, which were intractable due to their complexity and ambiguity.[10]

[10] You probably don't need to read original papers to know how these systems work. I would suggest looking at online resources detailing the history of symbolic AI.





If I know a robin is a bird, then I know it has wings

Semantic networks are knowledge structures that **efficiently** represent knowledge about the world

They can be used to make inferences about novel exemplars (generalise)

Collins & Quillian, 1968

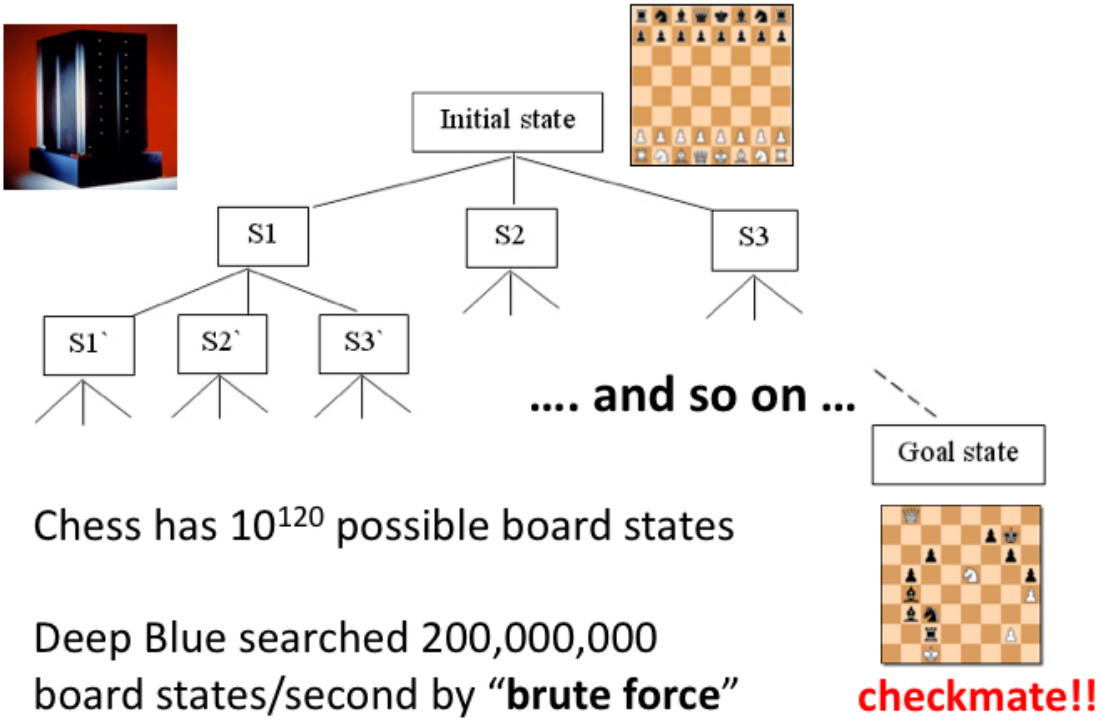
Another important advance in 20th century AI research was the development of "semantic networks"[11] which defined the structural relations among entities as a knowledge graph (see above). Like other GOFAI methods, the graph (once specified by the researcher) could be used to make new inferences. For example, if the graph specifies that a canary is a bird and birds have wings, it is possible to infer that canaries have wings, or even that a new item (e.g. a robin, connected to bird) also has wings.

[11] Rather than reading the original paper I advise looking at the summary in this article, which is discussed in detail in lecture 5: McClelland, J.L., McNaughton, B.L., and O'Reilly, R.C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychol Rev *102*, 419-457.

In 1997 IBM's Deep Blue, became the world's best chess player…

…by beating Gary Kasparov, the current world champion

If we fast forward to 1997, we come across one of most important successes in early AI research: the development of Deep Blue, which beat reigning world champion Gary Kasparov at chess. It had long been thought that the development of a machine that could beat a human grandmaster at chess would be the moment at which AI was "solved" (wrong. Now, you can see why defining intelligence is so critical to the project of AI research). Kasparov didn't like being beaten and claimed that IBM had "cheated".

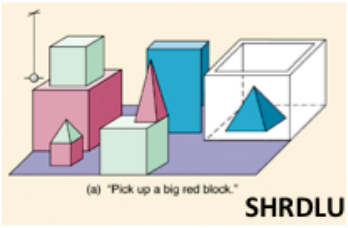


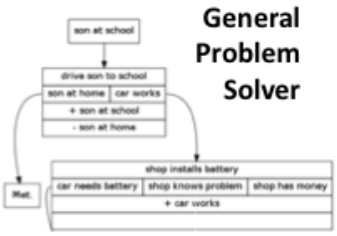
But IBM hadn't cheated. Rather, they had built a (then) powerful supercomputer, that was able to search through future board states at a (then) astonishingly fast rate – 200 million moves per second. This powerful computational tool allowed Deep Blue to estimate the potential future value of each different move it could make, using an efficient search method known as

"alpha-beta search". Deep Blue didn't learn anything about chess as it played, and it didn't find "patterns" in the board or use its intuition. It just searched for the best move by "brute force".

1.7. Critiques of the symbolic approach to AI

These systems can solve single complex problems, but they fall short of displaying **general intelligence**.





First, the systems are brittle (inflexible).

For example Deep Blue would fail catastrophically at Backgammon, or Checkers. This is because all computation is handcrafted by the researcher.

That's not cheating, but in a way, it's not all that smart either. In fact, what all of these GOFAI methods have in common is that they were tailored for a particular domain (a world of blocks, or the game of chess) and would fail catastrophically in any other problem they were faced with. Although the methods developed (e.g. alpha-beta search) have wider application, the AI itself was limited to solving one class of problem. That's very different to biological intelligence – whose hallmark is a flexible ability to solve whatever problem the agent comes across.

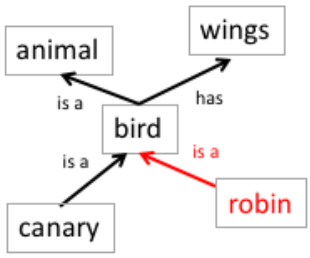
A useful set of definitions

Narrow intelligence: the ability to do a complex task well

General intelligence: the ability to do *any* complex task well

Artificial General Intelligence (AGI): the ability to do any task at least as well as a human

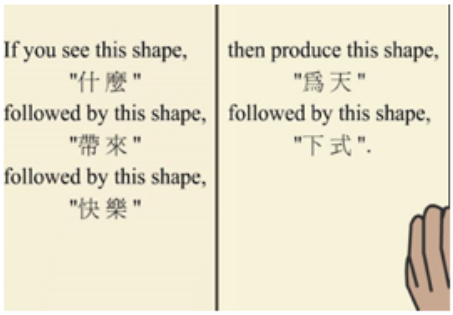
So is that "intelligent"? At this point, a set of definitions might be useful. Deep Blue and other GOFAI systems were intelligent, but their intelligence is *narrow*, that is, it is directed at solving a single problem. By contrast, today AI researchers are motivated by the dream of building a system with *general* intelligence, that is, the ability to perform well at any task. What does it mean to perform "well"? Of course, this is hard to define. But most researchers would be satisfied if they were able to build a system that could perform any new task at least as well as a human. This is the standard definition of "artificial general intelligence" (AGI), or as it is termed above "human-level machine intelligence" (HLMI).



semantic network

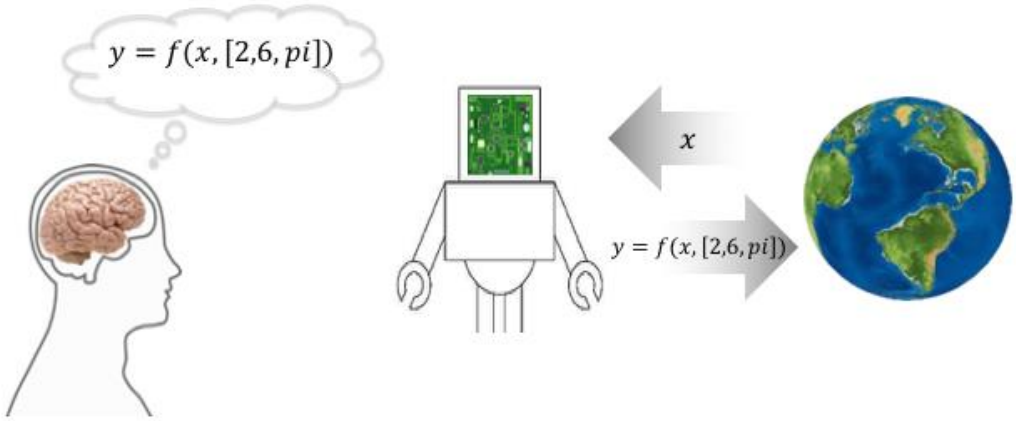
but this node only encodes "wings" because the researcher says so. The system doesn't "know" what wings are.

Searle's Chinese Room argument: the ability to manipulate symbols according to lawful principles doesn't imply understanding of their **meaning**



Harnad, 1999

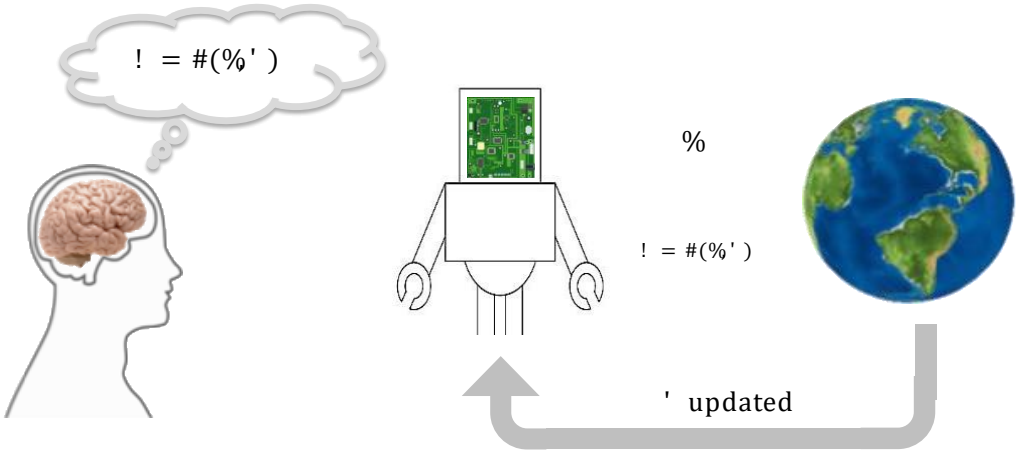
However, there is another, potentially more serious problem with these symbolic processing systems, and it's known as the *symbol grounding problem*. The problem is somewhat philosophical in nature and has its most eloquent exponent in the form of philosopher John Searle[12]. The problem is that there is nothing linking the symbols in the machine's memory to anything real in the external world. Take, for example, a semantic network. A node in the graph might correspond to "wings", but it only does so because the researcher has specified this. The network doesn't actually know anything about the real world. Why is this a problem? Well, John Searle offered a thought experiment that he called the "Chinese Room". Pieces of paper with writing in Chinese characters are passed into the room through a slit. Searle (who does not speak Chinese) sits in the room with a large book that specifies exhaustively the rules for mapping from the input Chinese characters to an appropriate response, also in Chinese. Searle copies the appropriate output onto the piece of paper and passes it back through the slot. From the perspective of someone outside of the room, the person in the room can "speak Chinese". But Searle doesn't understand Chinese at all. He's just transforming symbols according to a set of rules. The problem is akin to trying to learn a new language by perusing a dictionary written entirely in that language. You might exhaustively learn a mapping from words to definitions, but you won't know at all what they mean. Searle argued, thus, that symbolic processing systems don't really "understand" anything at all.



In traditional AI systems, both the function and the parameters (i.e. the agent's knowledge) are specified by the researcher

One of the reasons that symbolic systems suffer from the symbol grounding problem is that they don't acquire any new knowledge themselves from the world. In fact, according to our definition above, they either don't have any "parameters" at all, or the parameters are specified by the researcher. In other words, any knowledge that these systems have is downloaded from the researcher's brain and built into the system, rather than being acquired from interactions with the environment.

[12] https://arxiv.org/html/cs/9906002

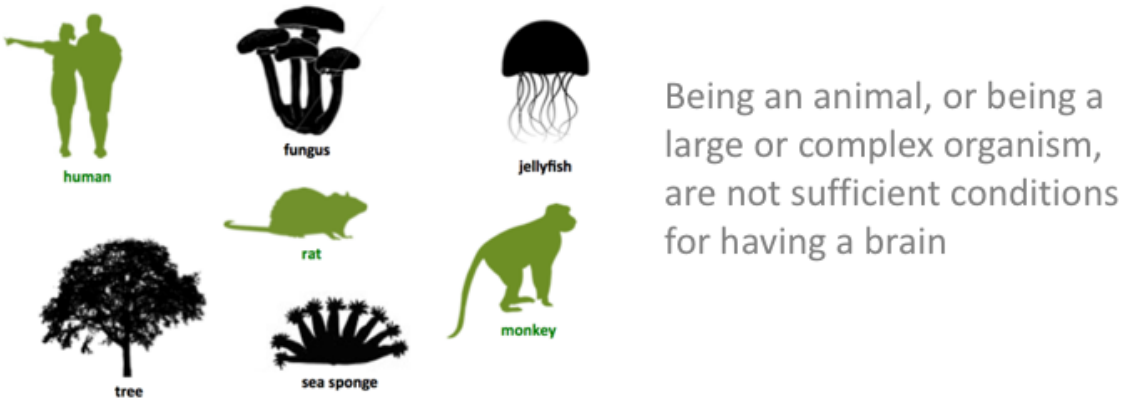


In machine learning systems, the parameters are learned (i.e. the system acquires knowledge) by experience, via an optimization principle

Contemporary machine learning systems, by contrast, acquire knowledge from the world. The parameters θ are typically initialised at random (like an infant, that begins by generating quasi-random actions) and gradually refined according to an optimisation principle (learning). In the next lecture, we will discuss reinforcement learning, arguably the most basic mechanism by which biological systems learn to select actions.

2. Reinforcement Learning

2.1. Why do we have a brain?



Brains are information processing systems that evolved to coordinate actions in response to complex sensory inputs

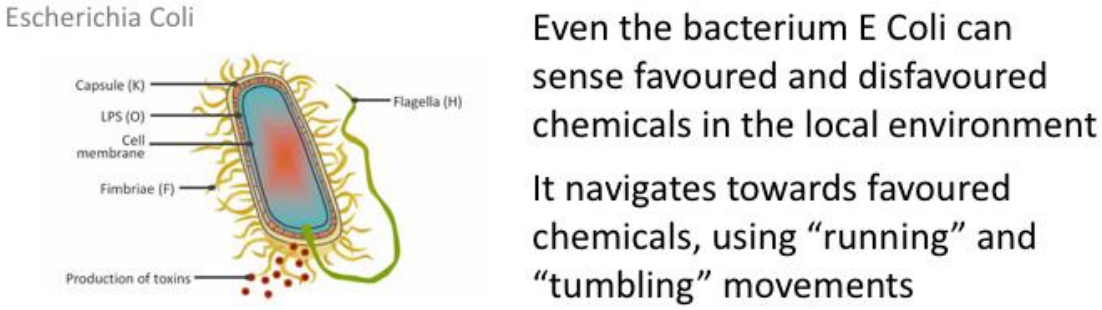
Why do we have a brain? If you ask most people (i.e. not psychologists or neuroscientists) this question (and I have) they mostly reply: "for thinking". Of course, they are right. But can we come up with a more precise answer to this question? In fact, there is some debate over this issue. But I'm going to outline the dominant view[13].

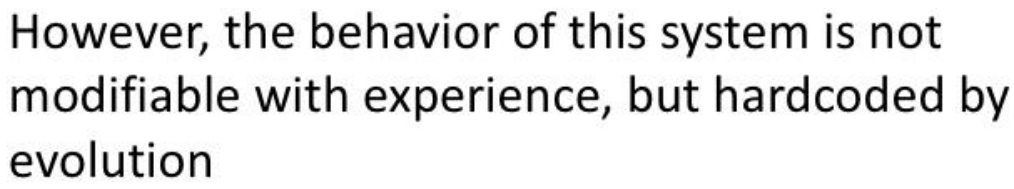
To consider why we have a brain, it's useful to think about which organisms have brains, and which don't. It might be tempting to assume that complex organisms have brains and simple ones don't. Clearly, however, this is not true. Trees are immensely complex organisms, but they don't have brains. An alternative suggestion might be that being an animal is a sufficient condition for having a brain. It's not. Sea sponges and jellyfish, for example, are animals but they don't have brains. If you look at the life forms shown in the figure above, only the ones in green have brains. What do they have in common?

The answer is that you typically have a brain when you need to flexibly select actions in response to external stimuli in order to survive. Jellyfish have a nervous system (they move through the sea) but they do so via a series of automatic reflexes which are fully determined by their sensory inputs. In the terms introduced in lecture 1, their brain is a function that has fixed parameters that cannot be modified by experience. Rodents, monkeys and humans however live in dynamic environments that constantly require them to decide among courses

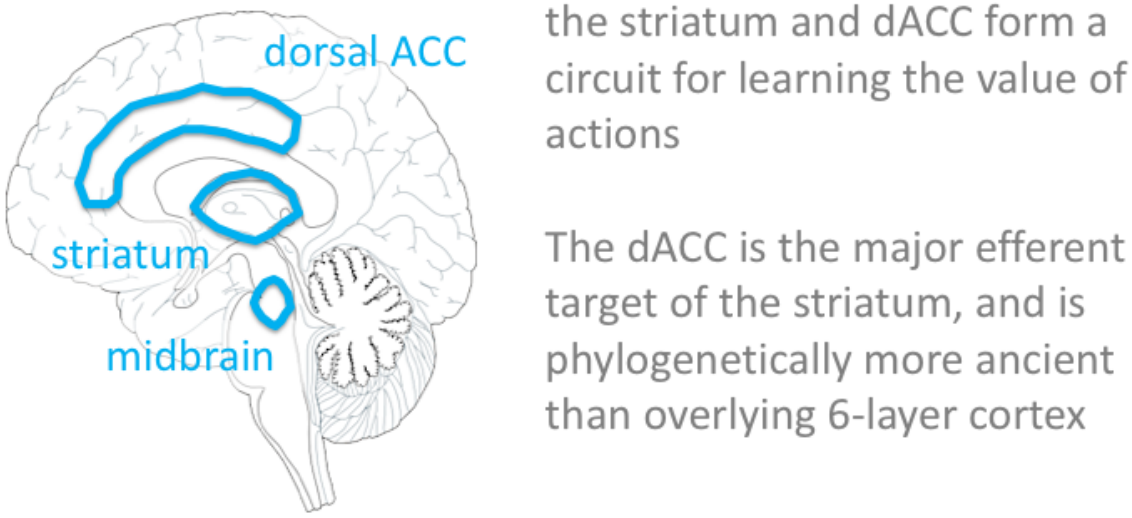
[13] For an alternative, this book about the octopus brain is great (but skip the bits about consciousness) Godfrey-Smith, P. (2016). Other Minds: The Octopus, The Sea, and the Deep Origins of Consciousness (Farrar, Straus and Giroux).

of action: which tunnel to run down, which branch to jump to, and which undergraduate degree course to follow. Brains are information processing systems that evolved to select actions in a flexible manner.



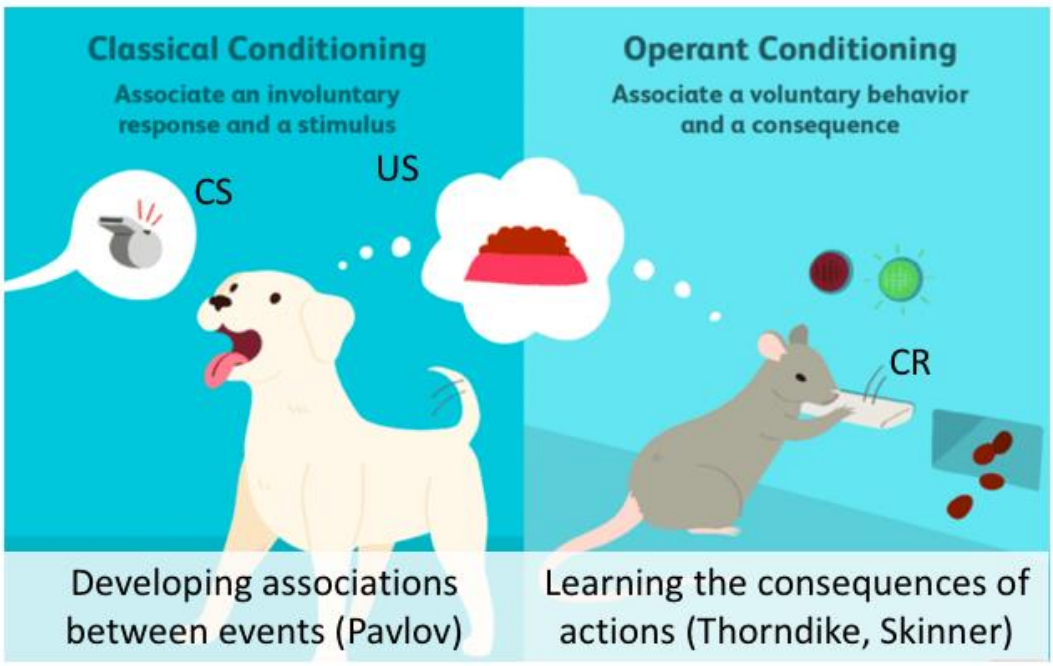


Just to really emphasise the importance of the need to select actions *flexibly*, action selection itself is not the province of organisms with brains. Consider this simple creature, E Coli (a bacterium that can cause nasty illness in humans). It navigates through the environment by following a chemical gradient, selecting "running" and "tumbling" movements in order to follow its desired trajectory. However, it doesn't have a brain. It can't – it's just a single-celled organism.



In this lecture, we're going to consider the function of two brain regions that are critical for action selection. The phylogenetically more ancient of these structures, the basal ganglia (BG), is among the most primitive brain structures, having retained the same morphology for at least 400 million years, and being present in our genetically most distant animal relatives, the lamprey eel and hagfish (slimy things that live at the bottom of seas/rivers)[14]. The BG receives inputs from the midbrain, which contains centres for the most basic functions (e.g. control of heartbeat, breathing, and gustation). The overlying dorsal anterior cingulate cortex (dACC) is an infragranular cortical area that can be thought of as an outpost of the basal ganglia that evolved subsequently (in mammals) just across the callossum from the BG. It performs a related but slightly more complex function, as we shall see below. In rodents, the dACC is a very prominent part of the frontal lobes; in primates somewhat less so; and in humans, it is rather dwarfed by the overlying granular (6-layer) cortex.

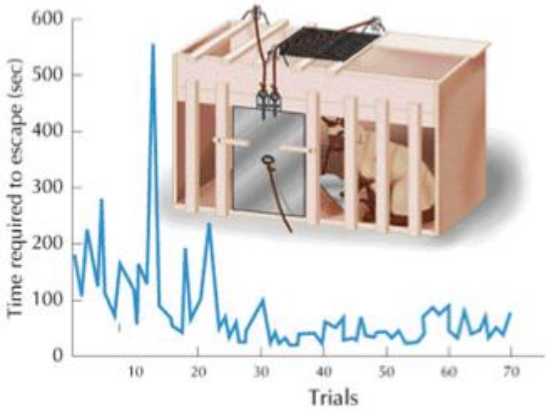
2.2. Classical and operant conditioning.



Classical and operant conditioning are the most basic forms of learning observed in biological brains. Classical conditioning (also known as Pavlovian conditioning after its discoverer, Ivan Pavlov) involves the learning of associations between sensory events. In the classic example, an unconditioned stimulus (US) such as food which naturally elicits an unconditioned response (UR; such as salivation) is paired by consistent temporal association with a conditioned stimulus (such a bell). Following learning, the bell will come to elicit the UR.

[14] Lots of interesting stuff about brain evolution in this book: Lynch, G. (2009). Big Brain: The Origins and Future of Human Intelligence (Palgrave Macmillan).

In operant conditioning (also known as instrumental conditioning) an animal learns that a particular action (conditioned response) such as a lever press elicits a stimulus, such as food. The *law of effect*, a term coined by Edward Thorndike, describes how the animal will come to produce the response more frequently when it elicits a (positive) stimulus, and less frequently when it elicits a negative stimulus (such as shock).



Actions that are rewarded will be repeated more often

Classic examples of operant conditioning include a cat who learned to repeat the actions that allowed it to escape from a box, and a pigeon being “shaped” by BF Skinner, as shown in the video.

Without further elaboration, the Rescorla-Wagner rule can account for operant learning:

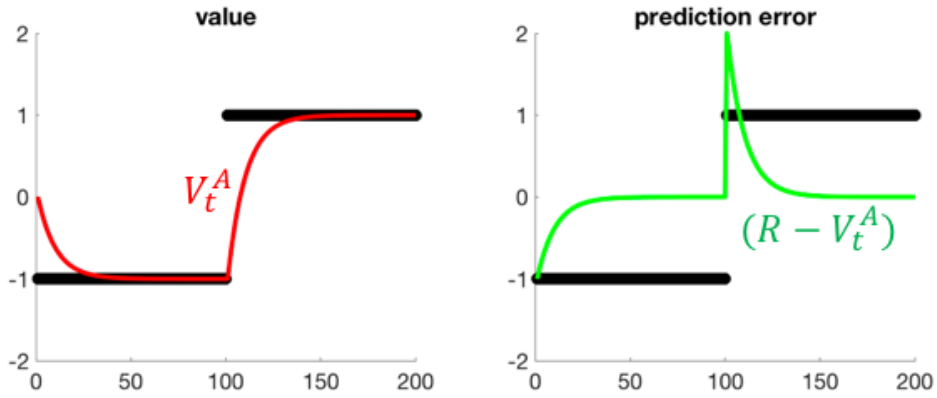
$$V_t^A = V_{t-1}^A + \alpha \cdot (R - V_{t-1}^A)$$

Translation: "The value of action A at time t, V_t^A is updated according its difference with the outcome R of action A, multiplied by some learning rate α"

This is sometimes also called a "delta rule model"

In the mid-20th century psychologists formulated a quantitative theory for understanding conditioning, known as the Rescorla-Wagner rule. It was originally proposed to explain classical conditioning but applies equally to the operant case[15]. The theory proposed that value of an action A at time t, denoted V_t^A was updated as a function of the discrepancy between the reward value it actually elicits R and a current expectation about the reward it will elicit, which is simply the value estimate from the previous trial V_{t-1}^A. This discrepancy, or prediction error, sometimes denoted δ, is scaled by a term α, known as the learning rate. The learning rate critically determines the extent to which the current update is influenced by the reward history. When the learning rate is low, the value estimate updates slowly, as a recency-weighted average of many outcomes associated with that action. When the learning rate is set to 1, the value of the action is simply set to R (in other words you think the action is good whenever it elicits a reward, and bad whenever it doesn't, irrespective of what happened in the past). This model is also sometimes known as a "delta-rule" model.

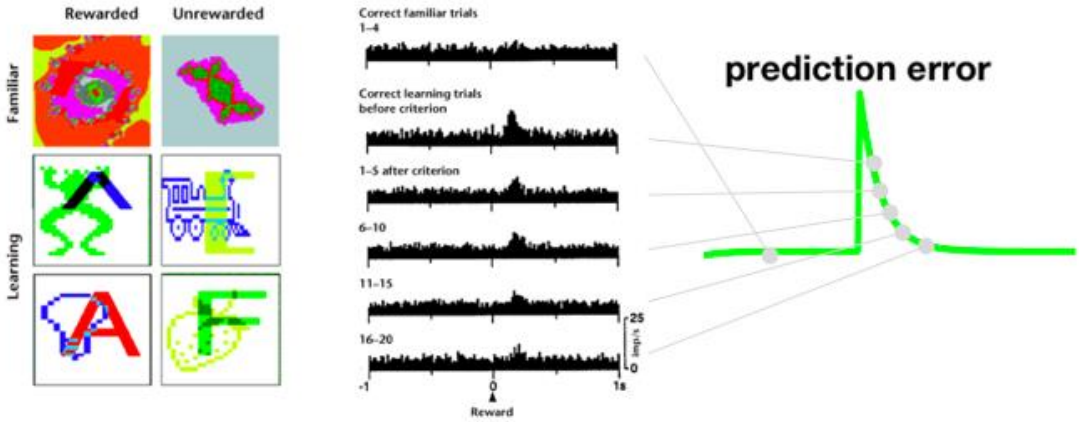
[15] The scholarpedia page on RW theory was written by Robert Rescorla himself. The notation he uses is slightly different to mine.



In this simulation, the model is applied to track a quantity that changes value after 100 cycles

$$V_t^A = V_{t-1}^A + \alpha \cdot (R - V_t^A)$$

To illustrate, here I plot the value estimate and prediction error for a simulated case in which a stimulus furnishes a reward of -1 on each of the first 100 trials, and +1 on the subsequent 100 trials. It is assumed that the model acts on every trial. The model initially has no knowledge of the value of taking an action and so V_0^A is initialised to zero. The value estimate converges to its true value (-1) after some tens of trials (here, the learning rate is set to 0.1; the rate of convergence would be faster if it were higher). Once it has reached -1, it does not deviate (at first) because the prediction and the reward match perfectly and so the prediction error is zero, and no further updating takes place. However, on trial 100, the reward unexpectedly switches to +1. This elicits a large, positive prediction error (spike in green trace) which gradually dies away as the model learns the new value of acting.

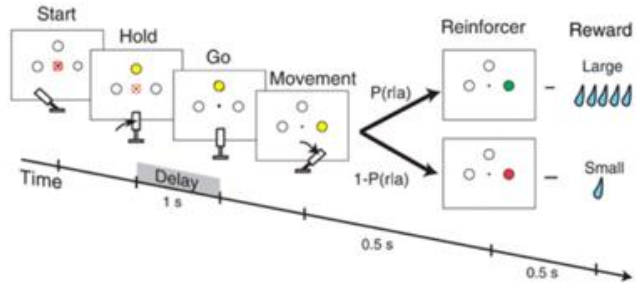


In an operant (visual discrimination) task, the responses of VTA dopamine cells to reward declined gradually during new learning

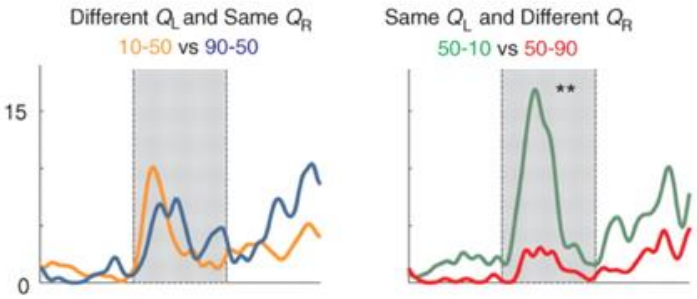
Hollerman & Schultz 1998

Let us compare to some empirical data[16]. These data were recorded from dopamine cells in a midbrain structure known as the ventral tegmental area (VTA) of the macaque monkey during a task that involved discrimination of two abstract images. First the monkey discriminates among 2 images with which it is highly familiar. Although it is rewarded on correct trials, the cells do not fire because there is no prediction error – the reward is fully expected. Subsequently, 2 new (previously unseen) images are introduced. The monkey at first doesn't know what to do. Correct responses elicit an unexpected reward, and thus a positive prediction error, and so the cells fire strongly when the reward is administered. As training proceeds, the monkey becomes more familiar with the stimuli and learns the correct response. Over time, thus, the neural signal elicited by the reward dies away, exactly in the manner predicted by the delta rule model in the previous slide. This study forms part of a large body of evidence, dating from the 1990s, which shows that VTA dopamine cells behave as if they were signalling reward prediction errors as computed by models of this class.

[16] Hollerman, J.R., and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. Nat Neurosci *1*, 304-309.



Monkeys made voluntary saccades to spatial locations to obtain rewards with varying probability



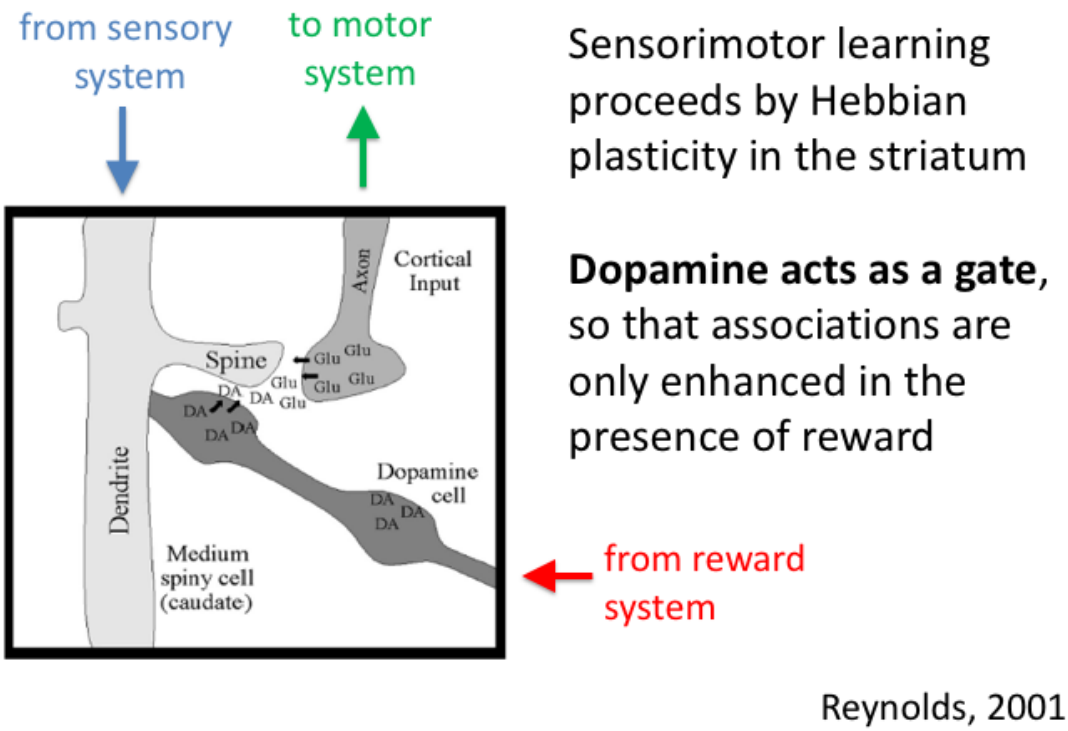
example neuron coding for V^{Right}

Cells in the striatum code for action values

Samejima et al 2005

Cells in the striatum (part of the BG) receive inputs from VTA dopamine cells. Further evidence suggests that some neurons within the striatum encode the value of particular courses of action. In this classic study by Samejima and colleagues[17], the monkey performs an instructed movement (left vs. right saccade) to a target to obtain a small or large reward (different volume of liquid). The experimenters varied the probability of reward for a left vs. right action. When the probability of a large reward conditioned on a rightwards response was 0.9, the cell in the lower panel responded vigorously, but it responded only weakly when that probability was 0.1 (right panel). However, when it was the value of responding left that was manipulated instead, the cell showed no difference in response. In other words, this cell is tracking the value of making a saccade to the right V_t^{right}.

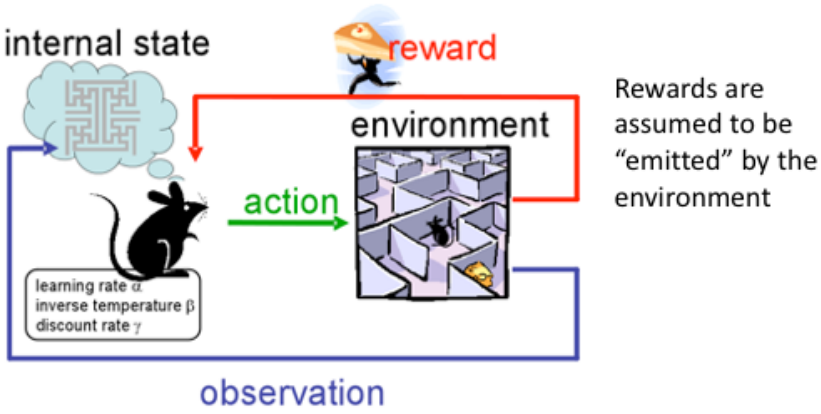
[17] Samejima, K., Ueda, Y., Doya, K., and Kimura, M. (2005). Representation of action-specific reward values in the striatum. Science *310*, 1337-1340.



So how is operant conditioning implemented at a neural level? The striatum is a critical brain structure for action selection. In human patients, lesions of the striatum lead to "akinetic mutism", a disorder in which patients become acutely apathetic and fail to select any actions at all, despite an apparent ability to do so. In addition to a dopaminergic input from the VTA (and substantia nigra), it receives inputs from, and projects back to, the cortex in partially-parallel loops that link sensory, motor and limbic zones. Consider an example striatal neuron that receives afferent inputs from a sensory region (coding, for example, the presence of a peanut) and projects back to the motor cortex, terminating on a motor neuron that codes (for example) for a reaching action. If the monkey sees a peanut and initiates a reaching movement, the connection between the input (sensory) neuron and striatal neuron will be strengthened by Hebbian learning (much more on this later), under the principle that neurons that are co-activated become connected more strongly. However, the critical algorithmic motif is that this update connection strength is modulated by (multiplied by) the level of dopamine that the cell receives from the reward system (e.g. VTA). Thus, if reaching for (and consuming) the peanut elicits a reward, there will be an update to V_t^{peanut} but otherwise there will not. This has been demonstated by measuring levels of synaptic potentiation following intra-cranial self-stimulation, a method that allows animals to press a lever to directly activate their own reward system (in this case the substantia nigra)[18].

2.3. Reinforcement learning and the Bellman equation

[18] Reynolds, J.N., Hyland, B.I., and Wickens, J.R. (2001). A cellular mechanism of reward-related learning. Nature *413*, 67-70. For a more detailed view: Wickens, J. (1993). A theory of the striatum (New York: Pergamon Press).



The framework of reinforcement learning describes how a system learns to select actions on the basis of observations and rewards

Sutton & Barto, 1998

This well-established biological framework for understanding how actions are selected has had a potent influence on the fields of AI and machine learning. The subdomain of reinforcement learning, which describes computational methods for choosing actions on the basis of the reward history, has become one of the dominant approaches with AI research. The currency of the RL framework is observations, actions and rewards. Observations are sensory signals emitted from the environment. The agent processes the sensory input, and selects and action according to a *value function*, which specifies the value of actions conditioned on the observation. The action leads to an outcome. The outcome is used to update the value function.[19]

[19] The definitive book on RL by Sutton & Barto was written in 1998 and updated in 2018. It is very long and you should not read it all! But various chapters may be useful. It can be downloaded here. For everything else that follows in RL, I strongly recommend chapter 4 of the Butz & Kutter book, which has really clear explanations.

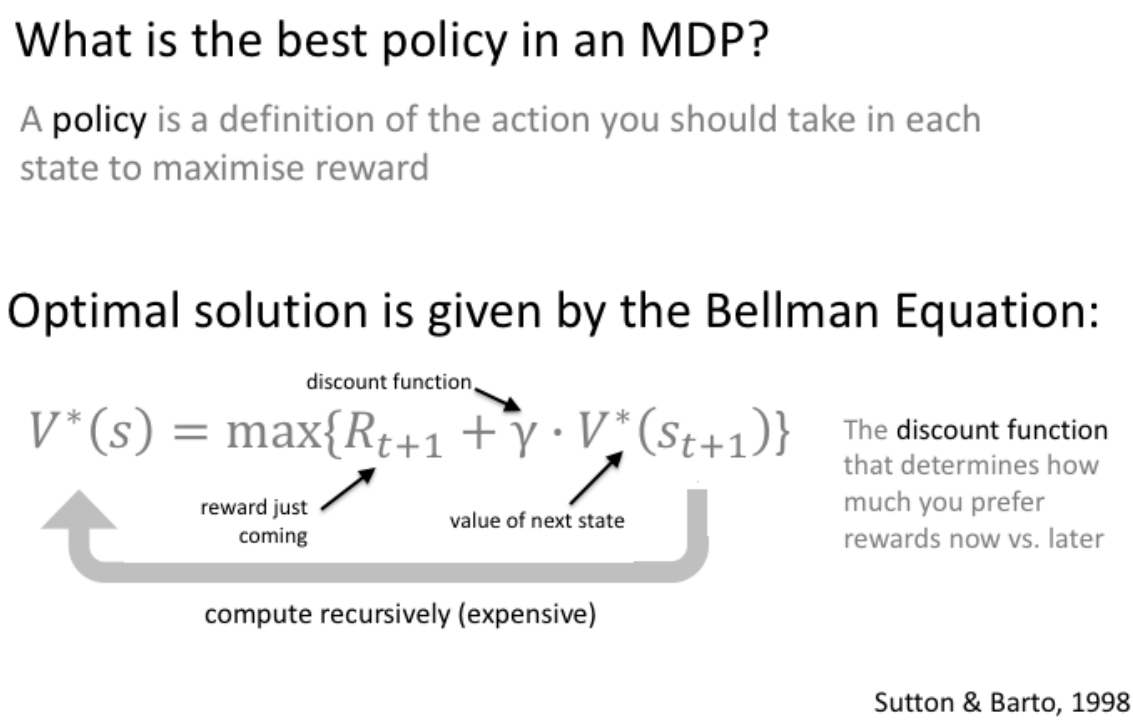
An MDP is defined by states **S**, actions **A**, a state transition function **P**, and rewards **R**

Markov property: future states s_{t+1} and rewards r_{t+1} depend only on current states s_t and actions a_t

Sutton & Barto, 1998

The widest application of RL methods is to a class of problem known as a Markov decision process (MDP). An MDP is an environment which can be defined in terms of a set of states S and their transitions P; a space of possible actions A; and the rewards that are likely to be obtained in each state R. An MDP obeys the Markov property, i.e. that future states depend only on the immediately past state. For example, in the grid world illustrated above, the state occupied by the agent (green) on the next step is fully determined by its current state and the action it takes – there is no need to consider past states (i.e. where it was before its current state) to predict its position.

The grid world problem illustrated is typical of the sort addressed in the domain of reinforcement learning. The agent occupies a start location which is distant from the goal. It has to learn to move directly to the goal to harvest a reward. How can it do that?



One advantage of working with MDPs is that it is possible to unambiguously specify the optimal policy for behaviour given S, P, A and R, thanks to work by the mid-20th century mathematician Richard Bellman. The Bellman equation is given above. The equation is used to compute how valuable you *should* think each state is, if you are an agent that is going to maximise reward in that environment. For example, in the grid world on the previous slide, an agent should think that it is better to occupy states close to the reward than far away. The Bellman equation looks complicated, but it's really very simple. It states that the optimal value of any given state $V^*(s)$ [i.e. the value of state s that you should compute in order to maximise reward in the environment] is obtained by maximising a quantity that is the sum of the reward obtained on the next timestep R_{t+1} and the discounted future reward that you can expect to obtain from that state. The discount factor ensures that expected rewards that might be obtained far in the future (e.g. £100 in a year's time) may be less subjectively valuable than those that are close to being obtained (e.g. £100 tomorrow).

The Bellman equation is simple to state, but it can be tricky to compute on the fly. That is because it involves *recursion*. To compute $V^*(s_{t+1})$ you need to estimate the likely value of $V^*(s_{t+2})$, and to know that, you need to estimate the likely value of $V^*(s_{t+3})$, etc. This can be computed using a set of computational tools known as dynamic programming, or alternatively can be approximated by sampling sequentially from possible future eventualities up to some reasonable time horizon. Luckily, however, a class of reinforcement learning algorithm that is closely related to the Rescorla-Wagner model, known as temporal difference (TD) learning, allows the optimal value function to be approximated by learning from reinforcement. This is the basis of contemporary reinforcement learning methods in AI/ML.

It is Monday. What is the value of the menu this week?

discounted future reward

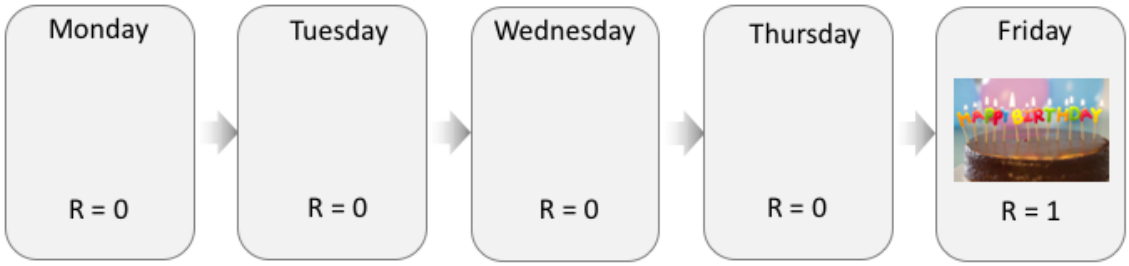
$$V^*(mon) = R(pizza) + \gamma \cdot R(salad) + \gamma^2 \cdot R(chicken) + \gamma^3 \cdot R(sushi) + \gamma^4 \cdot R(fish)$$

today's reward

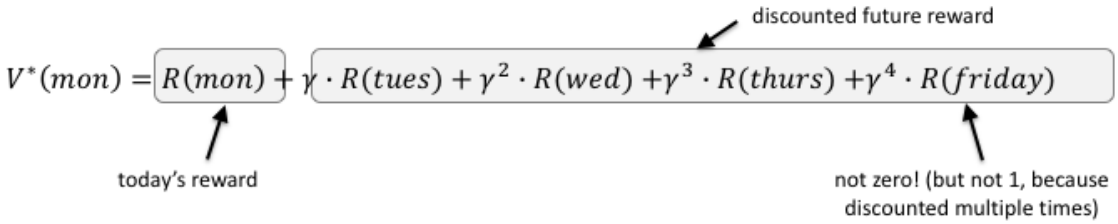
$0.9 \times 0.9 \times 0.9 \times 0.9 = small\ number$

Sutton & Barto, 1998

Let's go through an example to make sure that you understand how the Bellman equation works. The figure above illustrates an example lunch menu at your university canteen for the coming week. It's Monday. What's the (optimal) value of the current state, $V^*(mon)$? Lunch today is pizza, which you quite like (value = 0.7). But tomorrow it's salad (boring! value only 0.3). You are really looking forward to friday, however, because it's fish and chips! The Bellman equation states that $V^*(mon) = R(mon) + \lambda \cdot V(tues)$. So that's 0.7 + $\gamma \cdot V(tues)$. But what is $V(tues)$? That in turn depends on what is for lunch on wednesday, etc. The full equation for computing $V^*(mon)$ is given on the slide. Note that because the discount factor $0 < \lambda < 1$ is applied on every step, it gradually shrinks, so that states far in the future are more heavily discounted.



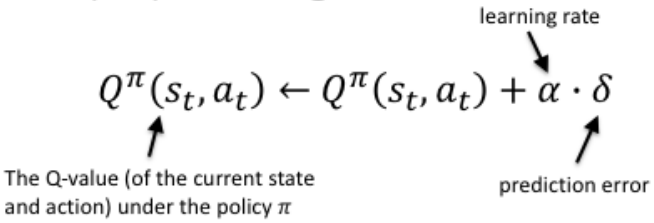
What about this week? Monday is boring, but on the other hand, your birthday is coming up.

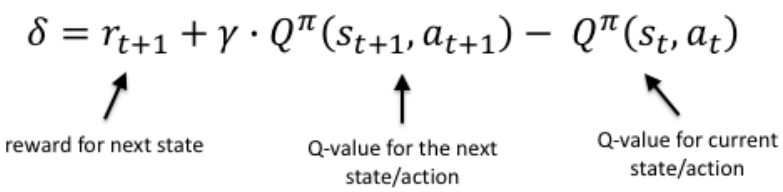


Here's another example that makes the link to our grid world problem more explicit. It's monday again, but the week is looking boring (canteen is closed). Except…it's your birthday on friday! So what's $V^*(mon)$? Well, you aren't getting any rewards on monday, but intuitively, it's not long to your birthday, so it's value probably shouldn't be zero. But it shouldn't be as good as (say) thursday. The equation shows why.

2.4. Temporal difference learning

We can approximate the optimal solution using temporal difference (TD) learning:





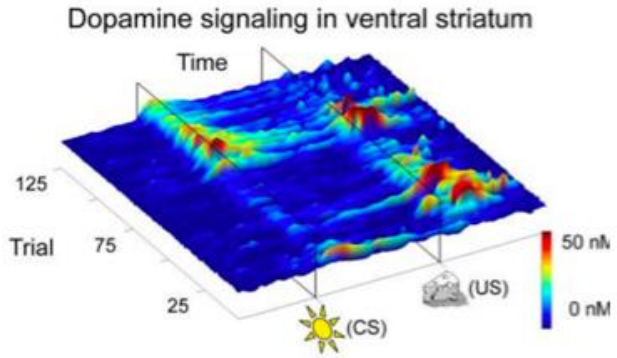
Note that the only addition to the delta rule is the new term for discounted reward of future action

Sutton & Barto, 1998

So how can we use TD learning to compute the optimal value function? To provide a more general solution, we're actually not going to compute the optimal value function V^* but a related function, known as the Q-function, which specifies not just the value of states but the value of each action in each state. So in the example involving the monkey and peanut above, we talked about learning V^{peanut} but actually here, we're going to learn $Q^{peanut,reach}$. Following the notation above, we're going to write the value of the current state and action under a given policy π (this is just the name given to the policy we are following, which in turn is determined by our Q-function) as $Q^{\pi}(s_t, a_t)$ where t is the current timestep. This is going to be updated by the prediction error δ multiplied by the learning rate α. Crucially, in TD learning, the prediction error is given by $r_{t+1} + \gamma \cdot Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)$.

So what does this actually mean? On each timestep, you have an estimate of the value of each course of action (initially, this is randomised, or set to zero). That's $Q^{\pi}(s_t, a_t)$. Now, take an action and add up the reward you are going to receive, and the value of the new state you are going to occupy. That's known as your "target". Subtract the estimate of the current state from your target, and that's your prediction error. Intuitively, you are comparing your current estimate of the value of that state to a new observed value estimate, i.e. expected to observed value estimate – exactly as you should.

Effectively, what TD learning is doing is gradually approximating the optimal value function for that environment under the Bellman Equation. The agent doesn't "think ahead", e.g. mentally simulate the trajectory from monday to Friday in our canteen example. Rather, the agent acts, and when a reward is encountered, it is "backed up" to the previous step. So, a reward obtained on Friday is used to adjust the value of Thursday, and then later, to update the value of Wednesday, and so on.



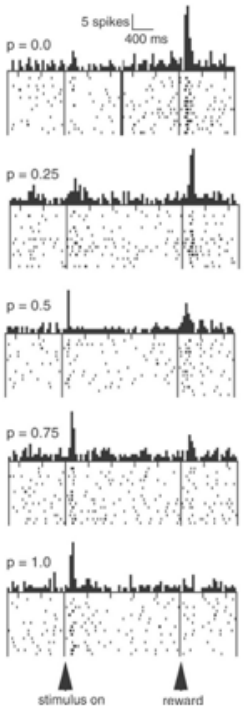
Dopamine concentration in ventral striatum measured with fast-scan cyclic voltammetry

Dopamine signaling in ventral striatum (nucleus accumbens) responds first to US, then (after training) to CS.

Flagel et al 2011

This should help us understand the well-known result that during classical conditioning, VTA dopamine neurons initially fire in response to the reward (which elicits a large prediction error), but this effect dies away over time. Critically, however the reward is then "backed up" to a stimulus that predicts the reward, earlier in time. Here, this is beautifully shown in recordings of dopamine concentration in the striatum using voltammetry[20]. Early in the block, dopamine is increased by onset of the US (i.e. the reward). However, over the course of training, the dopamine response shifts backwards in time, to the CS (e.g. a light that predicts the subsequent reward). Single-cell recordings from dopamine neurons in the VTA disclose a similar phenomenon.

[20] Flagel, S.B., Clark, J.J., Robinson, T.E., Mayo, L., Czuj, A., Willuhn, I., Akers, C.A., Clinton, S.M., Phillips, P.E., and Akil, H. (2011). A selective role for dopamine in stimulus-reward learning. Nature *469*, 53-57.



Dopamine neurons in VTA following onset of a reward-predictive stimulus and the receipt of reward

When the reward is unexpected, the neurons signal a reward prediction error (RPE)

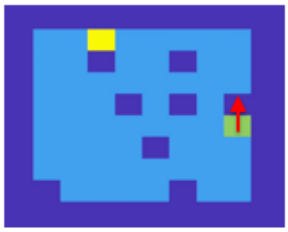
When the reward is expected, the signal shifts to the predictive cue

Fiorillo et al 2003

Another illustration is given in this study by Fiorillo and colleagues[21], who varied the predictive association between a cue and a reward. After training, when the cue typically signalled the likely absence of reward, a reward that did occur was very unexpected and elicited a large positive prediction error. When the cue was fully predictive of reward, the reward itself elicited no increase in firing rate, but instead the cue drove the dopamine neurons to fire vigorously.

[21] Fiorillo, C.D., Tobler, P.N., and Schultz, W. (2003). Discrete coding of reward probability and uncertainty by dopamine neurons. Science *299*, 1898-1902.

If the agent always chooses the most valuable action("greedy" choice) then it risks getting "stuck"



Imagine our agent had learned the following Q-values:

up	down	left	right
0.9	0.8	0.2	0.1

$argmax(Q(s_t, a))$ is 0.9, i.e. "up"

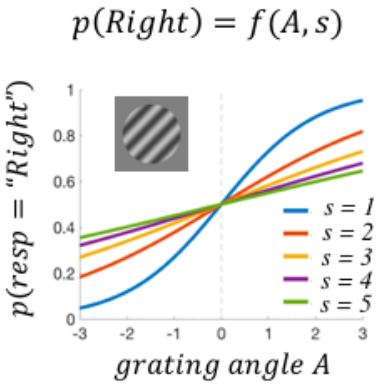
No movement = no learning, so it will try to go "up" forever (unless there is a cost for hitting the wall)

One solution: with probability ε, choose randomly, and with probability $1 - \varepsilon$, choose optimally

Watkins, 1990

The equations for TD learning we have discussed thus far describes how the value function should be updated. They do not, however, specify how the agent should act. Recall that the Q function encodes the value of each possible action in each possible state. So, in our grid world, the actions might be up, down, left or right. One obvious solution might be to always choose the best action. But this is poses a problem: imagine that the agent (green) occupies the position as shown, and for some random reason, the value of moving "up" is highest. The agent will be stuck in its current position, and because it's stuck, it can't learn anything new and thus remains stuck. More generally, we need a policy for action selection that ensures that some of the time, new courses of action (which may potentially be better) are explored. This is the well-known tradeoff between exploration (trying out a new course of action) and expoitation (choosing the option that you current think is best). We all experience this dilemma when visiting a familiar restaurant – the pizza was good last time, but what if the spaghetti is even better? You won't know unless you try.

So one solution to this problem is known as epsilon-greedy action selection. This simply says: on each timestep, behave randomly with probability epsilon (typically, small) otherwise choose the best option. This is quite a sensible policy. However, choosing randomly might be a bad idea in some circumstances, for example if you are next to a cliff.



Decisions made by animals are subject to intrinsic variability

Alternatively, choose action a_i in proportion to the value of all n action values, using a **softmax** rule:

$$p(a_i) = \frac{e^{Q(a_i)/\tau}}{\sum_1^n e^{Q(a)/\tau}}$$

Where τ is an inverse slope or **temperature** parameter (larger values of τ mean more random choices)

A more biologically plausible action selection policy is called a softmax. The equation is given on the figure above, but broadly, it means: choose actions roughly in proportion to how good they are. So if action A is twice as valuable as action B, choose action A approximately twice as often on average. Implementing this sort of policy gives rise to the sorts of behaviour observed in biological systems, where choice functions tend to be a approximately sigmoidal, which is the form of the cumulative response probability distribution given by the softmax rule. These considerations help us understand why biological systems make choices that are intrinsically variable, but broadly sensible.

2.5. Q-learning, eligibility and actor-critic methods

A similar method updates according to the value of the best possible (future) action, not that dictated by the current policy:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \delta$$

learning rate

The Q-value (of the current state and action)

prediction error

$$\delta = r_{t+1} + \gamma \cdot Q^{max}(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

reward for next state

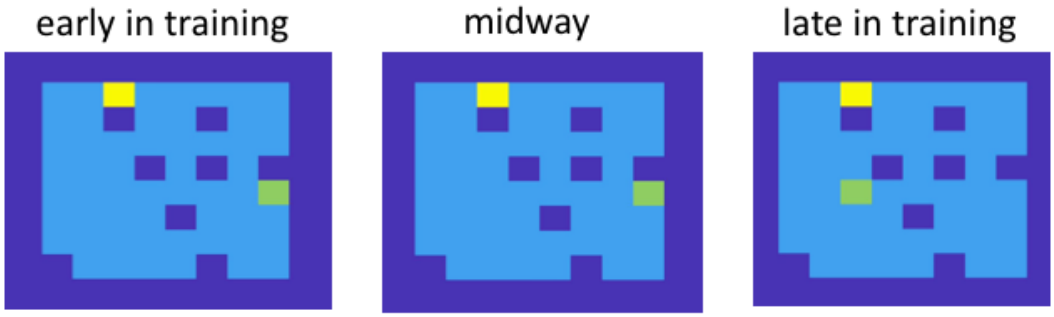
Best Q-value for the next state

Q-value for current state/action

Note that the only addition to the delta rule is the new term for discounted reward of future action

Watkins, 1990

TD learning is what is known as an "on policy" method. This means that the state estimation is carried out under the assumption that you are following the policy π that is given by your current value function. Of course, you might want to estimate the value of the next state under any policy that you could potentially follow. The best way to do this would be to estimate the value of the next state as $Q^{max}(s_{t+1}, a_{t+1})$, i.e. as the maximum value over all possible actions. This is known as "Q learning".

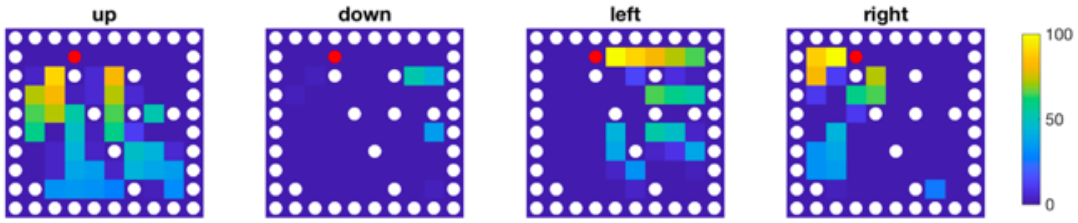


At first, the agent behaves randomly

By the end of training, it behaves optimally (makes a beeline for the goal, irrespective of the start location).

Sutton & Barto, 1998

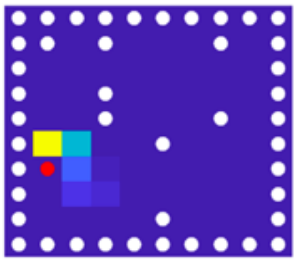
The videos in the figure above show the behaviour of an agent (green) trained to reach a goal (yellow) in a grid world. The three figures show its behaviour early, middle and late in training. As you can see, early on the agent behaves approximately randomly, whereas later it makes a beeline directly for the goal, because it has learned to approximate the optimal value function.



red dot is the goal location; white dots are the walls

Visualisation of the Q-values reveals the policy the agent has learned. The rewards are "backed up" from the goal to adjacent states.

Having trained our agent, one trick we can do is to examine its value function. Here, I've plotted the Q-values for the 4 actions: up, down, left right in each state. As you can see, when it's immediately to the left of the goal (now shown as a red dot) the value of right is high but left is low, and vice versa. The value of "up" is high in states that lead to those adjacent to the goal. Because the goal is at the top of the environment, there is rarely any benefit to going "down", and so Q values for down are all near to zero.

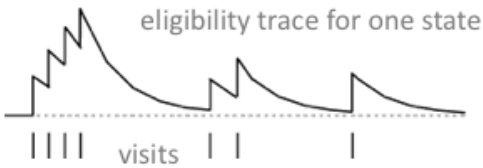


An eligibility trace allows the reward to be backed up faster through recently visited (or "eligible") states, rather than just the current state

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \delta \cdot E$$ ← eligibility trace

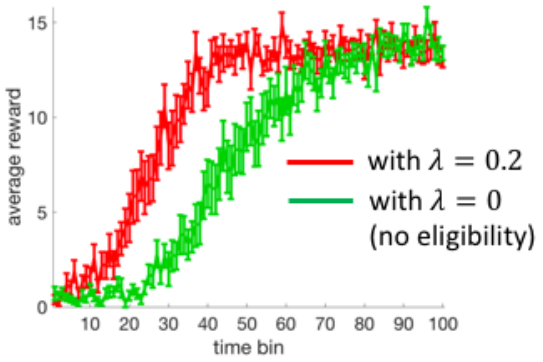
$$E(s) = E(s) \cdot \gamma \cdot \lambda$$ ← decay eligibility trace by lamba + discount (gamma)

$$E(s_t) = E(s_t) + 1$$ ← update **current** state by 1



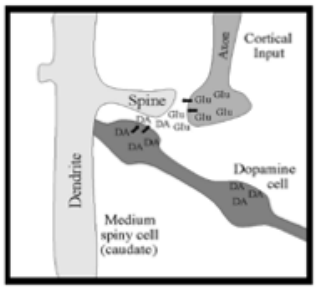
Sutton & Barto, 1998

One additional trick that we can do to speed up learning is to add an *eligibility trace*. An eligibility trace keeps track of how recently a state has been visited, by updating an eligibility function $E(s)$ each time it is occupied, and then gradually decaying the eligibility over time. At the time of update, δ is additionally multiplied by E, ensuring that values are not just backed up to the previous state, but to all states as a function of their recent visitation history.



The agent learns faster with an eligibility trace, because rewards are backed up more rapidly

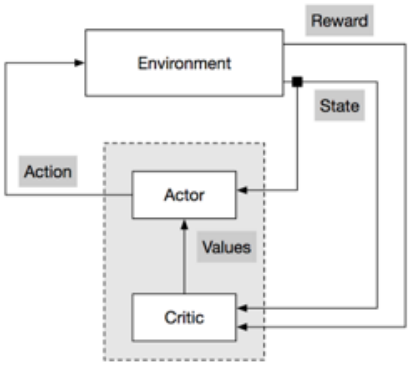
Neurally, eligibility traces may be mediated by slow timescales of learning or synaptic "tagging" at the synapse



See Gerstner (2018)

As you can see from the plot above, the addition of an eligibility trace accelerates learning (red) relative to the case without. At the neural level, it is not entirely clear what might constitute an eligibility trace, but there is evidence that synapses become "tagged" when stimulated, triggering a latent mechanism that may allow their later strengthening when a reward is received[22].

[22] https://arxiv.org/abs/1707.04192



actor (direct policy)

$$\pi(s_t|\theta) \leftarrow \pi\,(s_t|\theta) \cdot \beta \cdot \delta$$

critic

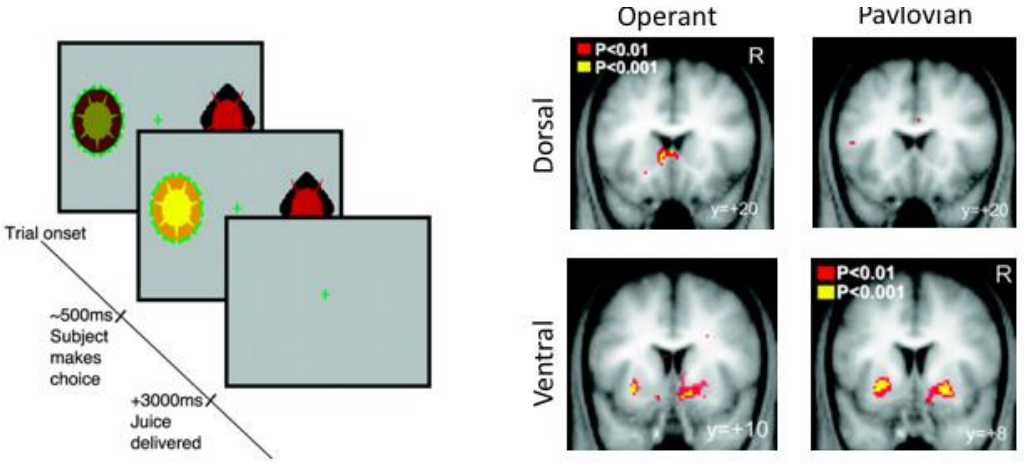
$$V(s_t\,) \leftarrow V(s_t\,) + \alpha \cdot \delta$$

update

$$\delta = R_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t, a_t)$$

The actor-critic model learns a separate state value function (V) and state/action value function (Q)

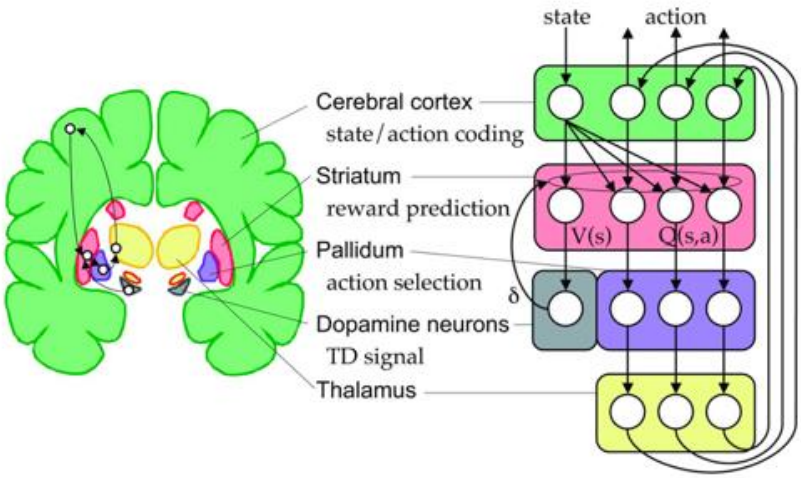
Finally, it is worth drawing your attention to a slightly different (and very successful) reinforcement learning approach, known as the actor-critic model. In actor-critic methods, one part of the system directly optimises the policy (with some parameters θ), i.e. what do; another learns the value of specific states. This seems to accord well with the neural separation between learning the value of stimuli (e.g. via classical conditioning) and the value of actions conditioned on a state (e.g. in instrumental conditioning).



fMRI reveals prediction errors in dorsal and ventral striatum during operant (e.g. actor) and Pavlovian conditioning (e.g. critic) respectively

O'Doherty et al 2003

Indeed, despite the shared computational principles, there is good evidence that classical and instrumental conditioning rely on different neural structures. For example, in this study by O'Doherty and colleagues[23], they measured neural prediction errors (in BOLD signals) in a design that allowed them to dissociate Pavlovian and operant learning. Operant conditioning elicited prediction errors in BOLD in both the dorsal and ventral striatum (nucleus accumbens), whereas those observed in Pavlovian conditioning were limited to the accumbens. This is consistent with the greater involvement of dorsal (rather than ventral) striatum in motor control. Note that the correlation between neural activity and prediction errors in the striatum seems like it contradicts evidence from single-cell recordings which has pointed to the midbrain (e.g. VTA) as the origin of dopamine prediction error signals. However, it is likely that BOLD signals measure principally afferent (input) activity to a region, which may explain the discrepancy between results from the two classes of recording method.



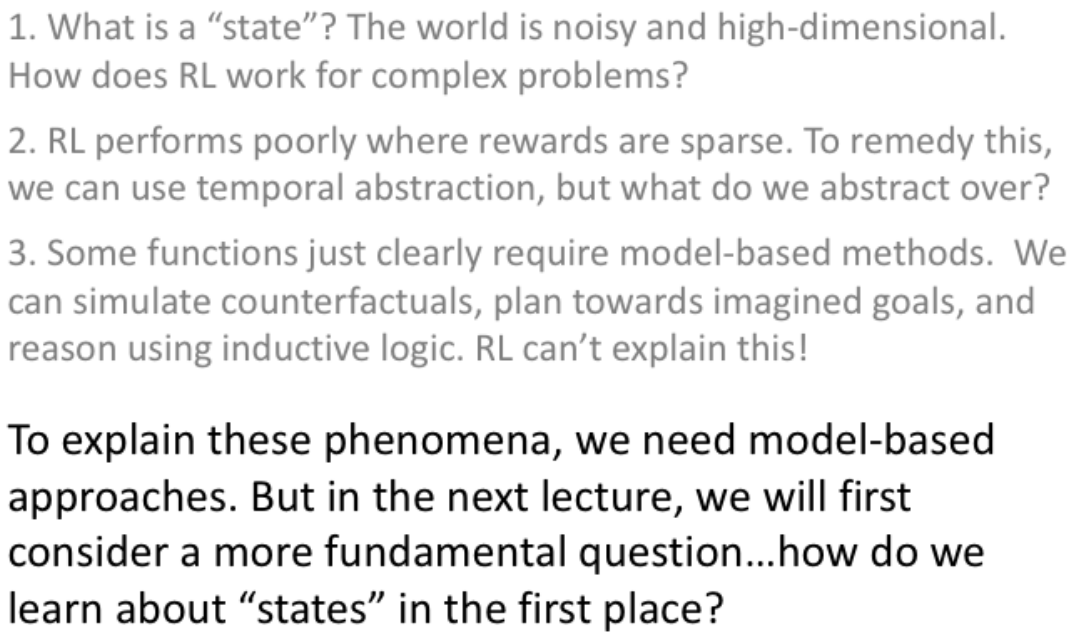
General models proposed for mapping computations of RL onto circuitry of basal ganglia

Doya, 2007

So now we are in a position to put everything together. Prediction error signals are computed in dopamine neurons of the midbrain, which send signals to the striatum. These gate the stimulus-stimulus (ventral striatum; V-function) and stimulus-response (dorsal and ventral striatum; Q-function) links formed during ongoing experience and behaviour, that depend on inputs to the striatum from the neocortex. The outputs of these systems are routed via the globus pallidus and striatum to the cortex, where they are converted into motor behaviour. Under this scheme, there is a nice correspondence between the reinforcement learning methods (which we know approximate optimal value learning, and work in practice) and the

[23] O'Doherty, J.P., Dayan, P., Friston, K., Critchley, H., and Dolan, R.J. (2003). Temporal difference models and reward-related learning in the human brain. Neuron *38*, 329-337.

functional neuroanatomy of the cortico-striato-cortical loops that are ubiquitously observed in the brains of both simple and complex animals.



However, there remain a number of limitations with the RL framework. Firstly, model-free RL methods (as discussed thus far) require that an agent learn a function that specifies the value of each action in each state. But what is a “state”? This is easy to define in a simple MDP like a grid world, but much harder to specify in the real world. Secondly, we know that model-free RL perform poorly large or complex in environments with sparse rewards. As we will see later, temporal abstraction can help, but how to find the right abstractions remains an unsolved problem. Finally, if we think about the richness of human action selection, it extends way beyond what is described here. For example, humans can simulate imagined or counterfactual goals, and select actions that lead to them. You couldn’t use RL to decide what career to follow or which house to buy! Clearly, there are other methods that are required to build truly intelligent action selection. Of which more in subsequent lectures.

3.Deep learning in the primate ventral stream

3.1. Parametric models for object recognition

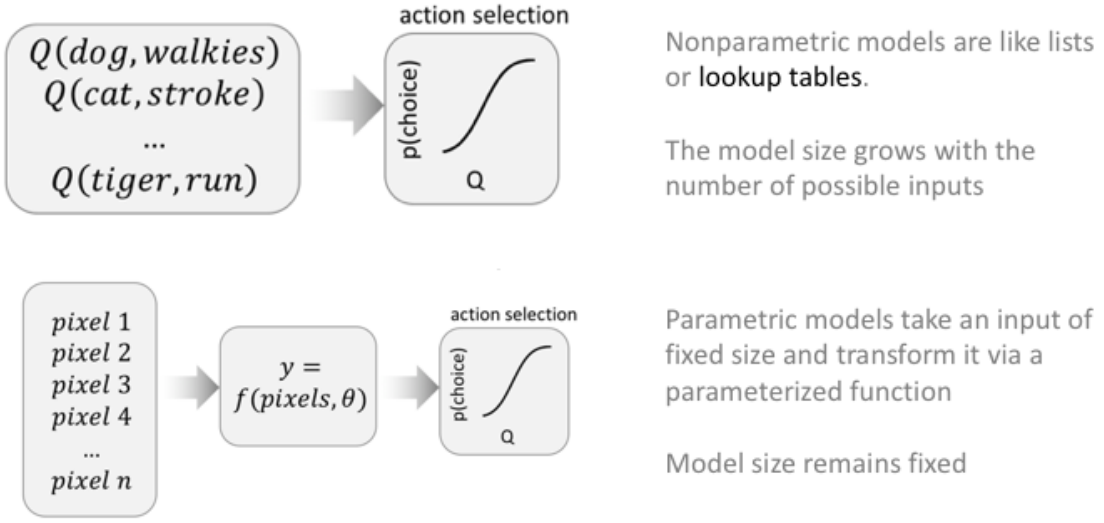
This is a cat

This is a dog

But how does the brain work that out?

The challenge of vision is the infinite variety of the universe. Every image (state) that is incident on the retina is at least partly unique!

The truly astonishing thing about the mammalian visual system is its capacity to engage in accurate object recognition despite the infinite variety in sensory stimulation. You have probably never seen the cat or dog shown above, but you have no hesitation in identifying them. Every single image that is incident on the retina throughout your lifetime is unique, but visual recognition (at least under photopic conditions) seems utterly effortless. How does the brain do that?

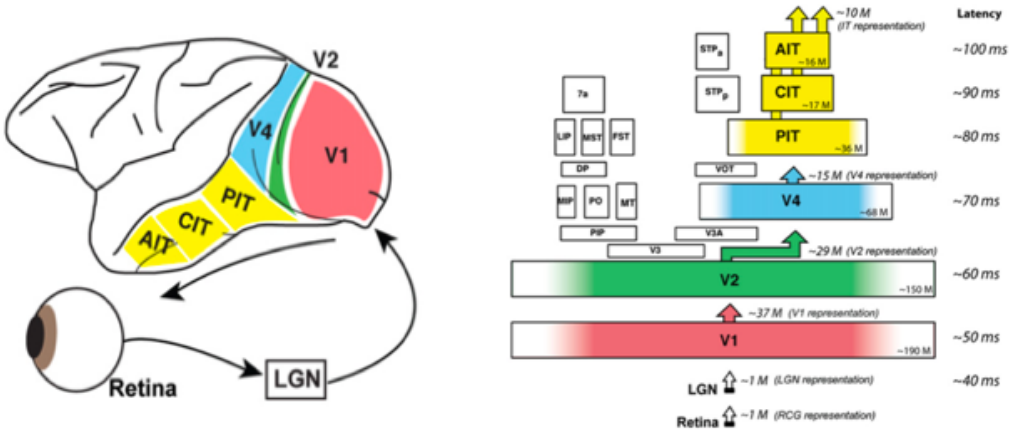


To deal with the infinite variety of the sensory world, we need a parametric model

In the previous lecture, we discussed RL models, that learned a value function encoding the value of actions in given states. But the infinite variety of sensory experience means that there are no uniquely specified "states" available to most animals, at least not that resemble the clear [x,y] locations in the grid world examples we used previously.

To understand how this challenge can be met, it's useful to draw a distinction between parametric and nonparametric models. Many standard RL models, such as the TD and Q-learning examples we discussed, are *nonparametric* models. This means that they encode values in something like a tabular format. One corollary of this is that as the number of potential inputs (states) or outputs (actions) grows, so does the size of the network (i.e. table). In a world in which each state is unique, this will not suffice – because we would need a new entry in the table for every new input that impinges on our sensory systems!

One alternative is to use a *parametric* model. In parametric models, the number of inputs and outputs is fixed. The model learns a set of parameters that map inputs onto outputs. Similar states will be passed through the network in comparable ways and produce similar outputs. In other words, the model will be able to *generalise* existing knowledge to new exemplars. Thus, you might respond to any new cat as you have responded to the cats you have previously met. Parametric models can in principle handle any number of distinct inputs, although in practice the number of distinctions it will be able to make among them is going to be limited by the number of parameters (i.e. the network size). In this lecture, we will discuss feedforward neural networks, which are a popular type of parametric model (or *function approximator*) that are often used for object recognition. Later in the course, we will see how parametric models can be combined with the RL framework to produce powerful RL agents that can behave intelligently in complex, open-ended environments such as video games.



During the forward sweep of visual processing (~100ms), a signal of dimensionality ~37M in V1 is compressed into a signal of dimensionality ~10M

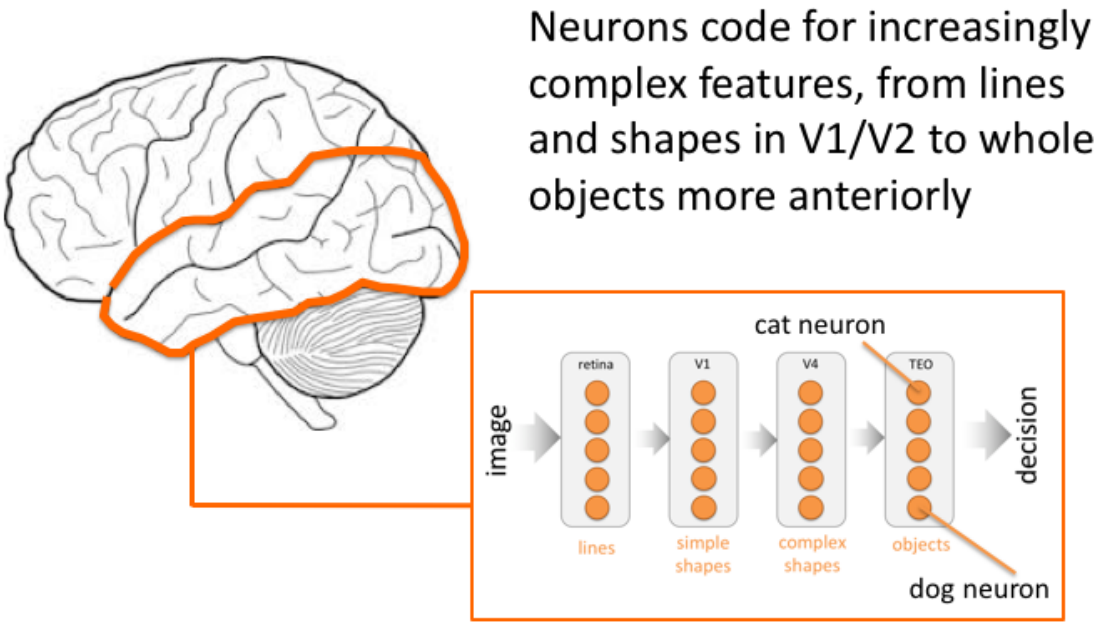
Let us begin by taking a close look at sensory systems. Sensory systems are most extensively evolved in mammals, and in humans and other primates, vision is by far the dominant modality. In this lecture, thus, we will focus on vision and use the terms “sensory” and “visual” in an interchangeable fashion (with apologies to colleagues who work on other fascinating sensory modalities).

Before diving into the details, it is probably worth saying a word or two about what sensory systems are for. Sensory systems, such as the primate ventral visual stream, allow inputs to be pre-processed so that they are in a state more suitable for cognition and action selection. In the images shown on the previous slide, from the point of view of the visual system the sensory inputs are not “cat” and “dog” but rather are complex, high dimensional data structures that encode the luminance and wavelength of each part of the image that is incident on a photoreceptor. The job of sensory systems is to reduce this complex data structure to a neural code that is readily interpretable by other systems that may wish to select appropriate actions to it – for example, stroking the cat or taking the dog for a walk. Although you may effortlessly identify a cat or a dog in the image, it is worth pointing out that this is not because object recognition is a trivially simple computational problem. Quite the converse. It’s easy because you are a primate, and primates – after millions of years of evolution – are very good at it.

It’s probably worth pointing out that some other animals get by quite well with much more primitive image preprocessing systems. For example, frogs have “bug detector” cells on their retina that directly code the presence of a small dot that enters the receptive field and stops or moves intermittently. Activation of these cells is sufficient to provoke a reflex tongue protrusion towards the appropriate spatial location. There’s no preprocessing of the image – retinal stimulation leads directly to a (predatory) action.

In the primate, the visual system consists of multiple hierarchically organised regions. During visual processing, information is passed forward in an initial sweep lasting ~100ms, from the

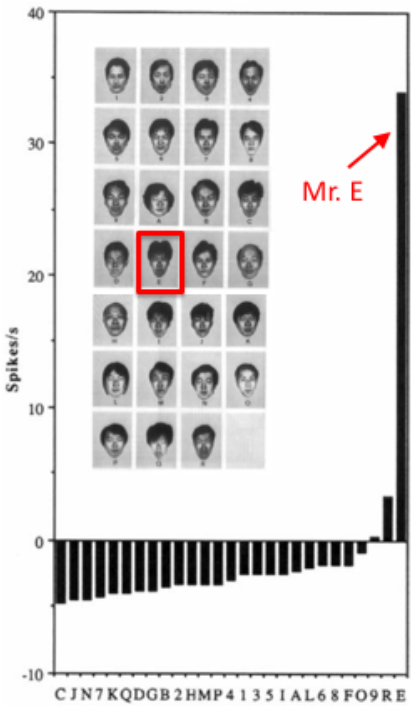
retina to the earliest stages (e.g. V1) and from there through the ventral stream to a plethora of extrastriate regions, terminating in anterior temporal lobe regions that lie close to the hippocampus and surrounding medial temporal lobe structures. During this feedforward processing, the dimensionality of the signal is reduced from ~37M (in V1) to ~10M (in AIT), although the effective compression that occurs between these stages (i.e. after taking neural correlations into account) may be far greater.[24]



Along the ventral stream, we know that receptive fields grow in both space and time (here, we shall focus on the expansion in space, leaving time for the next lecture) and that the complexity of RF response properties grows. Thus, a cartoon description of the representational properties of the primate ventral stream might state that early visual neurons code for simple visual features, such as orientation and spatial frequency, whereas later visual neurons code for complex objects, faces and locations, including for example putative neurons coding for "cat", "dog" "your grandmother" in area IT/TEO.

Neuroscientists have long sought a unified theory that can account for the diverse and complex (but seemingly highly organised) coding properties of visual neurons. However, in order to formulate such a theory, one has to define exactly what those coding properties are in the first place. This presents a major challenge in itself.

[24] DiCarlo, J.J., Zoccolan, D., and Rust, N.C. (2012). How does the brain solve visual object recognition? Ibid. *73*, 415-434. See also Kriegeskorte, N. (2015). Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. Annu Rev Vis Sci *1*, 417-446. And Yamins, D.L., and DiCarlo, J.J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nat Neurosci *19*, 356-365.



View dating back to the 1960s: sparse coding of shapes and objects in extrastriate visual cortex

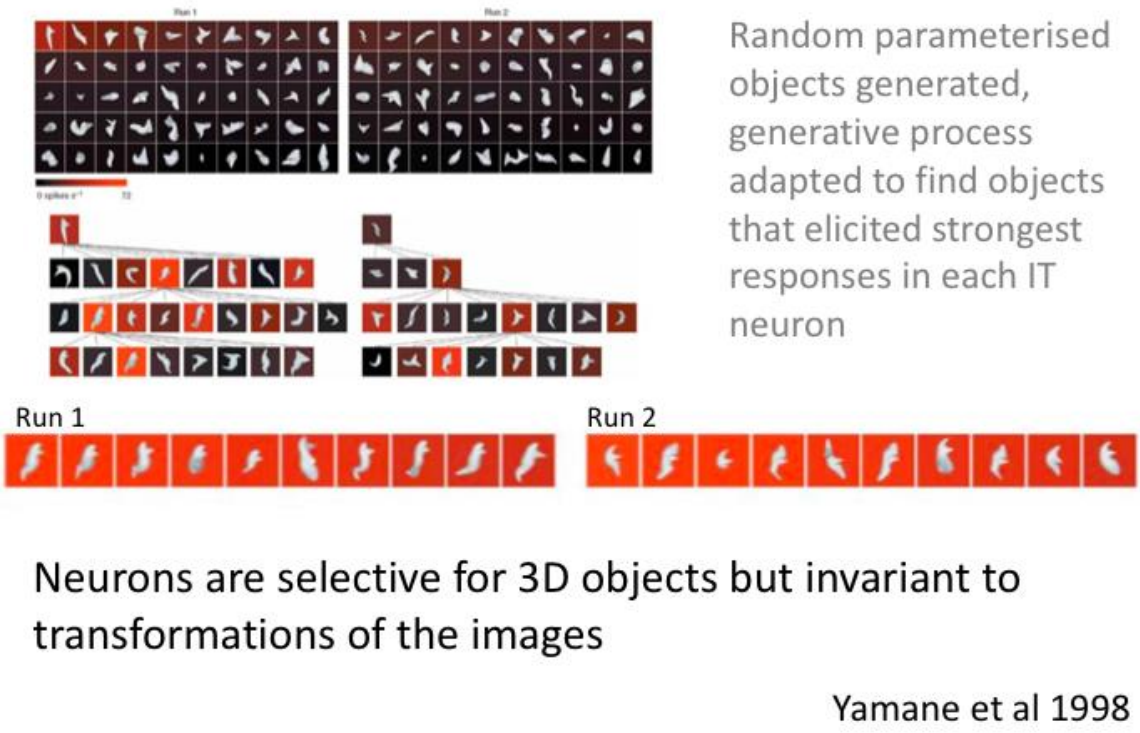
i.e. there are "grandmother cells" that code for a specific object identity

However, most cortical coding is distributed (see Bowers, 2009 for a useful discussion)

Young & Yamane 1992

This slide highlights an early finding suggestive of the fact that there may be very "sparse" representations in the anterior temporal lobe, i.e. cells that respond to a very specific object or, in this case, individual. The researchers presented macaque monkeys with images of human faces, and the slide depicts a neuron that responded to a single person and barely at all to everyone else. Lo, a grandmother neuron! However, despite the pleasing story that this tells (and the surprising vindication of a theory that began as a joke in the 1960s), we now know that most coding in the anterior ventral stream is quite distributed, i.e. many neurons each code for many features. Estimating the true sparsity of coding in any recording region is of course challenging, given the (very) limited sample of neurons that can be recorded out of the many millions that may be present.[25]

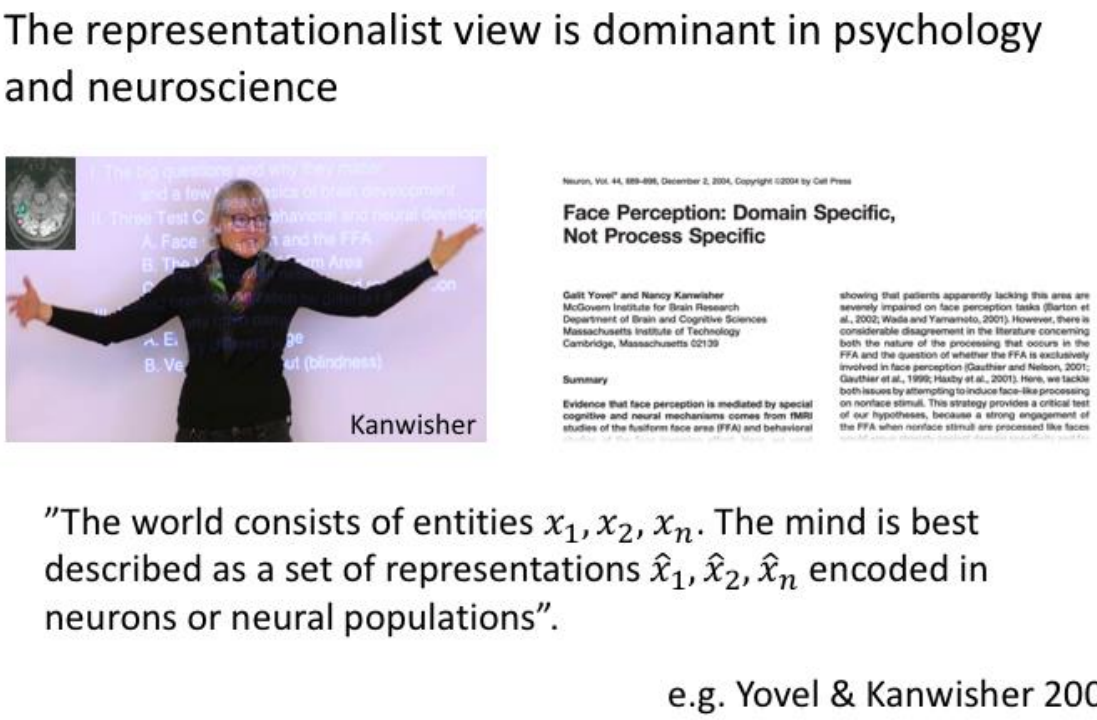
[25] Primary paper is this one. Young, M.P., and Yamane, S. (1992). Sparse population coding of faces in the inferotemporal cortex. Science *256*, 1327-1331. See also this review by Bowers Bowers, J.S. (2009). On the biological plausibility of grandmother cells: implications for neural network theories in psychology and neuroscience. Psychol Rev *116*, 220-251. See also this article from the Quiroga (the Jennifer Anniston neuron guy): Quiroga, R.Q., Kreiman, G., Koch, C., and Fried, I. (2008). Sparse but not 'grandmother-cell' coding in the medial temporal lobe. Trends Cogn Sci *12*, 87-91.



That same problem – the limited sample of neurons that can be recorded in any one study – makes identifying the specific coding properties of visual neurons (or, any neuron) very complicated. However, the authors of this study identified a very clever way of visualising the neural code in IT[26]. They presented monkeys with random 3D objects that were generated with a combinatorial description language, that allowed them to generate a virtually limitless set of stimuli. Whilst recording from each neuron, they iterated the stimulus parameters through multiple generations in a way that was optimised to maximise the response of the neuron. For example, if a cell responded to stimulus A, they presented more stimuli that resembled A; if it then responded most to A_1, they presented yet more stimuli that resembled A_1. The continued this process until they had identified a family of stimuli to which the cell responded extremely well. Two examples are shown on the slide above. The cells responded to 3D objects with a well-defined structural form but were invariant to rotations and translations of those objects.

3.2. A critique of pure representationalism

[26] Yamane, Y., Carlson, E.T., Bowman, K.C., Wang, Z., and Connor, C.E. (2008). A neural code for three-dimensional object shape in macaque inferotemporal cortex. Nat Neurosci *11*, 1352-1360.

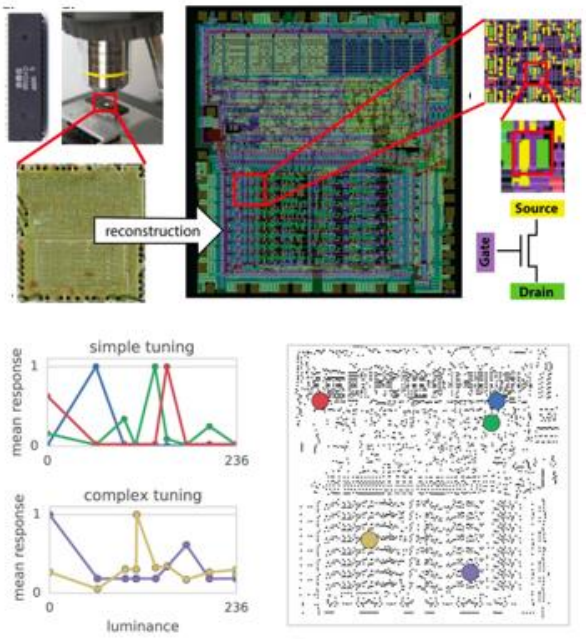


At this point it may be useful to take a small digression and discuss the dominant mode of understanding of the sensory systems of biological brains, and where its limits lie. A long tradition in psychology and neuroscience is based on *representationalism*, that is, the philosophy that a brain can be understood by exhaustively identifying the coding properties of its constituent neurons or brain areas. This has not been a fruitless endeavour. Much of our foundational understanding of how neural systems function comes from recording experiments that have defined their coding properties, such as Hubel and Wiesel's seminal experiments in cat V1. However, it is sometimes forgotten that identifying representational properties is only useful as a means of pinpointing the *computational* principles by which the system works – of specifying how stimulus(Kanwisher, 2017) information is transduced from one format into another and ultimately guides behaviour.

Arguments over "what a neural signal codes for" can rapidly become sterile in the absence of any wider theory about what that region computes. A paradigmatic example of the intellectual cul-de-sac that a pure representationalist stance can lead to is the debate over the "function" (read: primary coding axis) of the "fusiform face area", a portion of the human ventral stream that responds more robustly to images of faces that other objects in fMRI studies. A number of research groups (and one in particular[27]) argued forcefully over a period of about 10 years that the "function" of this region was to code for faces, rather than to engage in more domain-general processing. Whilst it is ultimately true that both cells and BOLD responses in this region seem to respond more vigorously to faces than to a range of other objects irrespective of the task context, in this debate it was largely overlooked that the "function" of any region is not to code for something. Rather, its coding properties are a by-product of the computations that it carries out; and ultimately the only substantive question is what a neural circuit computes, not what it codes for. The goal of any mature theory, must thus be not simply to assert what the

[27] Let's be fair and give Nancy Kanwisher the right of reply: you can see her perspective on this episode in the field's history here: Kanwisher, N. (2017). The Quest for the FFA and Where It Led. J Neurosci *37*, 1056-1061.

coding properties of a region are; it must be to define the computational principles by which representations are formed.



Take a system that is fully understood because we built it (e.g. a microprocessor)

do cognitive neuroscience e.g. single cell recordings, brain imaging, lesion studies

Results reveals tuning curves, connectivity profiles, lesion-symptom maps and oscillatory activity just as in the real brain. **But we know that the interpretative logic applied to these phenomena is completely wrong!**

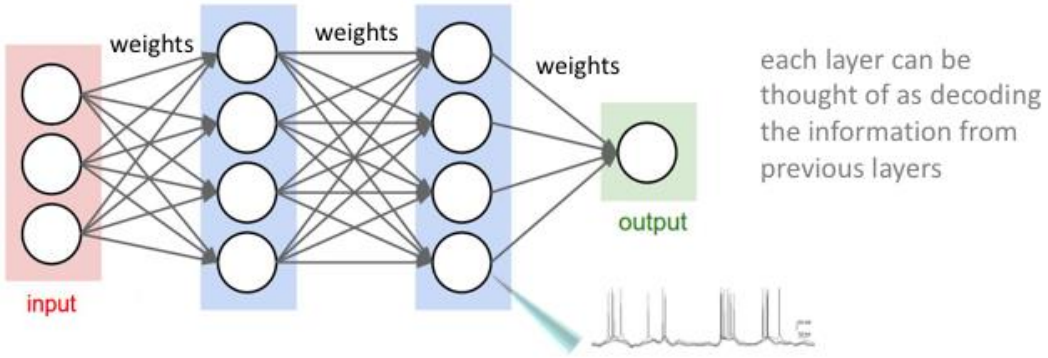
Jonas & Kording 2017

The limits of the pure representationalist approach were beautifully laid bare by this extremely clever paper from the group of Konrad Kording[28]. Building on a historical argument, the authors set out to highlight the limits of the interpretative logic that we habitually employ when conducting empirical investigations. Neuroscientists study a system they do not fully understand – biological brains – and on the basis of their findings, they make inferences about the functioning of the system according to an interpretative consensus in the field. For example, if a lesion to a region impairs a particular function, we typically assume that that region is causally involved in that function; if a cell responds to a given experimental variable, we typically assume that its job is to code for that variable.

In their paper, the authors asked: if we take a system we *do* understand fully (because we built it), conduct comparable experiments, and apply the same interpretative logic, will be recover theories about the functioning of the system that are in accord with the (known) ground truth? The answer was firmly "no". The authors conducted "experiments" on the microprocessor used to control the screen pixels in an Atari video game, and carried out the equivalent of the "single cell recordings", "lesions" and "connectivity analyses" typically employed by neuroscientists. Their data showed many of the stereotyped phenomena that are typically observed in neuroscience experiments (e.g. chips had "tuning curves"). However, we know that the consensus interpretative logic applied was entirely misplaced, because the conclusions that a neuroscientist would naturally draw from the data were simply incorrect (e.g. the chips were not encoding information from the screen). This demonstrates, more than

[28] Jonas, E., and Kording, K.P. (2017). Could a Neuroscientist Understand a Microprocessor? PLoS Comput Biol *13*, e1005268.

any other paper, the caution that must be applied when trying to understand what a neural system "represents" in the absence of a plausible theory about what it is computing and why.



When we measure tuning properties, we are decoding the information that is sent to a neuron, which in turn depends on its connections (input weights)

So a different way to understand a neural system is: what is the principle by which the connections are learned?

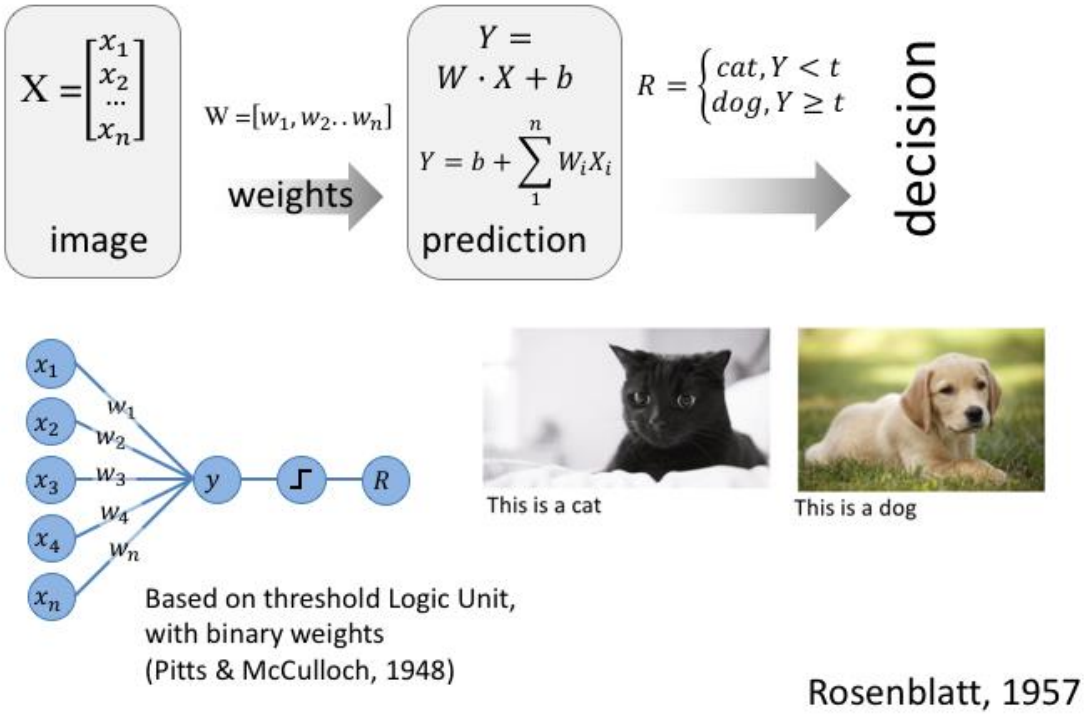
So, as we have discussed, neuroscientists like to talk in terms of the coding properties of cells in the brain, and their goal is to formulate a theory of how those coding properties arise. In trying to provide a wider answer to this question, as we will do here, it is worth beginning by contemplating how those coding properties arise in the first place. Neurons have receptive fields, that is, the portion of the external world to which they are sensitive. They also have tuning properties, that is, a function that describes how they respond to various parameters – such as the tilt of a grating or the frequency of a sound - that the experimenter has manipulated (or otherwise thinks are important for the function of the network in which the cell participates). But where do these coding properties come from?

Listening to the extreme representationalist position taken by some researchers, it might be tempting to assume that these coding properties are endowed by the organism's genetic heritage, i.e. each cell's tuning sensitivity is hardwired by evolution, which has chosen to furnish V1 with edge detectors and IT with grandmother neurons. However, there is an alternative to this argument. To understand this alternative, it is important to realise that any cell's coding properties are a direct function of its inputs. Thus, a cell has a receptive field at a given location of retinal space because its inputs can be traced back to those photoreceptors that lie at precisely that position on the retina. Thus, ultimately it is the patterns of neural connectivity that determine the coding properties of all cells. In the language of machine learning, these patterns of connectivity arise because of the optimisation principle that dictates how network weights (or synaptic strengths) are updated as a function of experience.

In this lecture, we will discuss a new proposal, which is that "deep" neural networks offer a computational theory that jointly explains the representational properties of neurons in the primate ventral stream, as well as how the system allows animals to recognise objects. We will

go on to offer a critique of this theory. But first, we will begin by explaining what neural networks are and how they work.

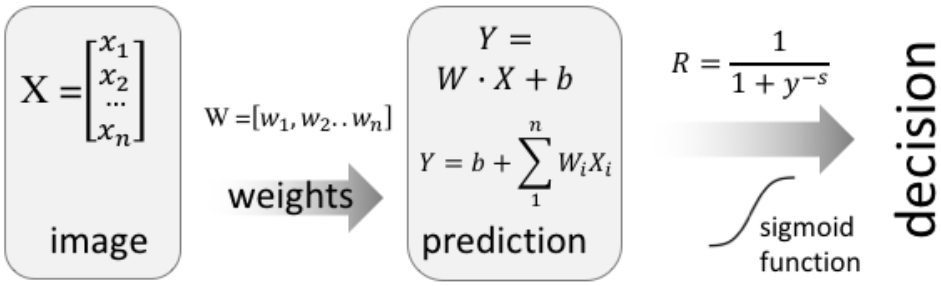
3.3. Perceptrons and sigmoid neurons



The invention of the computational tools from which neural networks are built predates even the Dartmouth conference at which the field of AI was officially inaugurated. The precursor to all neural networks is arguably the threshold logic unit (TLU), first developed by Pitts and McCulloch in the 1940s. The TLU is a computational component that combines binary inputs and then applies a threshold to generate a logical output [0,1]. In the TLU, the weights are handcrafted by the researcher, in the spirit of the GOFAI approach that sought to build powerful logical processing systems from a set of simple computational components.

The perceptron was a successor to the TLU[29]. The perceptron differs from the TLU in a number of ways, but for our purposes the most important is that the network is able to learn by itself, as weights are updated by a delta-rule like mechanism. Note that today the term "perceptron" is often used when referring to a sigmoid neuron (see below), which is historically inaccurate.

[29] This page explains the difference most clearly: http://ecee.colorado.edu/~ecen4831/lectures/NNet2.html



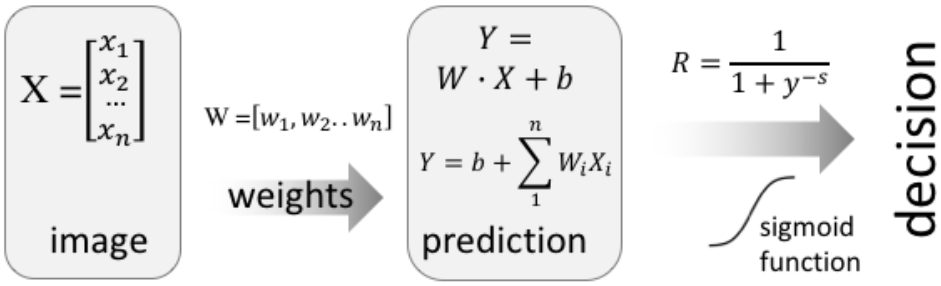
Supervised learning relies on a teaching signal T that denotes the ground truth (was it a cat or a dog?)

Note that this assumes an external oracle (or supervisor) that knows the true object class

However, it is the next generation of network components – the sigmoid neuron – that is the most important precursor to contemporary neural networks. The sigmoid neuron takes real-valued inputs and computes their weighted sum Y. Rather than being converted to a binary output via a hard threshold, Y is passed through a sigmoid function (we have encountered this function above; it maps any number onto a value in the range [0,1] via an ogival curve). Below we shall discuss why this latter feature is important for learning.

You may have noted the similarity between the computation conducted by a sigmoid neuron and multiple regression. You can think of the inputs X as the predictor matrix and the output Y as the dependent measure; the weights W are the regression coefficients. In fact, with the inclusion of the sigmoid (logistic) function, you can think of a sigmoid neuron as implementing multiple logistic regression in an online fashion.

The key feature of this class of network is that it is trained with *supervision*. A supervision signal indicates the "true" answer to the problem the network is trying to solve, as if provided by an external oracle or teacher. For example, let us imagine that the goal of the network is to classify a vector of inputs (say, the pixel luminance values in an image) as +1 (=cat) or -1 (=dog). On each trial, the network sees an image and outputs +1 or -1. Our optimisation principle aims to adjust the weights so that the network outputs a 1 whenever an input corresponds to a cat and -1 when the image is a dog. This is what we mean by "learning" in a supervised neural network.



The learning rule for a perceptron is very simple:

$$W = W + \Delta W \quad \text{where} \quad \Delta W = X \cdot \frac{1}{2n}(T - Y)^2 \cdot \alpha$$

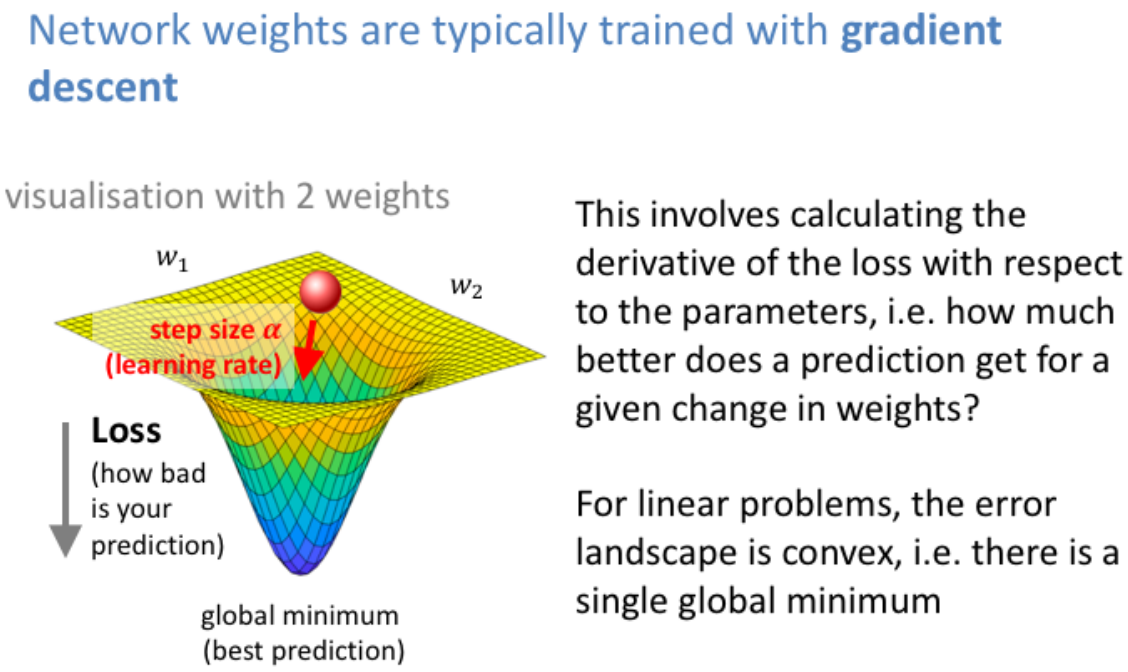
This updates the weights according to the gradient (rate of change) of the loss (mean squared error) with respect to the parameters

*the scaling $\frac{1}{2n}$ is just a convenience

Most neural networks of this class are trained using *gradient descent*. On timestep zero, one would first initialise the weights at random, and pass the inputs through the network. In a sigmoid neuron, this would entail computing the output as $R = \sigma(W \cdot X + b)$ where X is the input vector, b is an additive bias term, and $\sigma(\cdot)$ denotes the sigmoidal transform. Let us say that the first image vector was actually a cat (i.e. +1) and the network output a value of 0.3 (the weights are random, so this could be anything). To train the network, we need to specify a loss term (or loss matrix) that is, to state how bad any given outcome is. For example, we can define our loss as the (squared) discrepancy between +1 and 0.3, or $0.7^2 = 0.49$.

The next step is to update the weights so that the loss is likely to be smaller on the next timestep – i.e. so our predictions are slightly more accurate. This can be done adjusting the weights according to their *derivative* (or rate of change) with respect to the loss. In other words, one can calculate how the loss is changing as the weights change and adjust the weights so that they are likely to minimise the loss in the future. This is the optimisation principle that underlies learning in neural networks. To do this, in a sigmoid neuron, we update the weights as the outer product of the inputs X and the loss, multiplied by a learning rate. The learning rate serves the same purpose as in RL models.

Critically, learning by gradient descent is only possible when a small change in the weights leads to a small change in the output value in one direction or another (i.e. greater or smaller). This means that how "wrong" the network was (i.e. how large the discrepancy is between the observed and desired state of the network) varies smoothly with the values of the weights. This is not possible when the output changes abruptly between 1 and 0, as in networks with logical outputs.



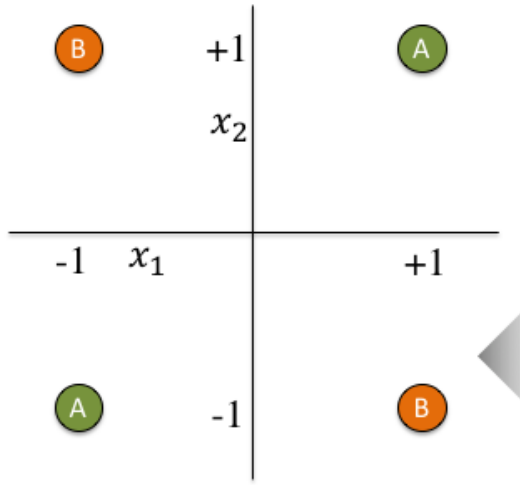
To understand how gradient descent works, it is useful to visualise the loss as a function of different values that the weights can take. In the example above, we imagine that there are just 2 weights (for ease of visualisation). There is a global minimum to the loss, that is a setting for w_1 and w_2 where the loss is minimal. If this value is close to zero, then the network has converged and its predictions will be accurate. You can think of the process of gradient descent as a ball rolling down this loss landscape to the minimum. The rate at which the ball falls will be given by the learning rate. If the learning rate is too large, the ball will miss the global minimum and jump back up the other side of the valley, even if it is heading in the right direction. This is why we often need a low learning rate when conducting gradient descent.

For linear problems, i.e. those that can be solved by a sigmoid neuron, the landscape is convex. That means that there is a single global minimum, and so gradient descent should always find the best setting for the weights. However, for more interesting nonlinear problems (e.g. cat vs. dog classification from image pixels), this is not always the case.

3.4. Depth: the multilayer perceptron

$$X^A = \begin{bmatrix} 1, -1 \\ 1, -1 \end{bmatrix} \quad X^B = \begin{bmatrix} -1\ 1 \\ 1, -1 \end{bmatrix}$$

Rows are different features (e.g. x_1) and columns are different exemplars

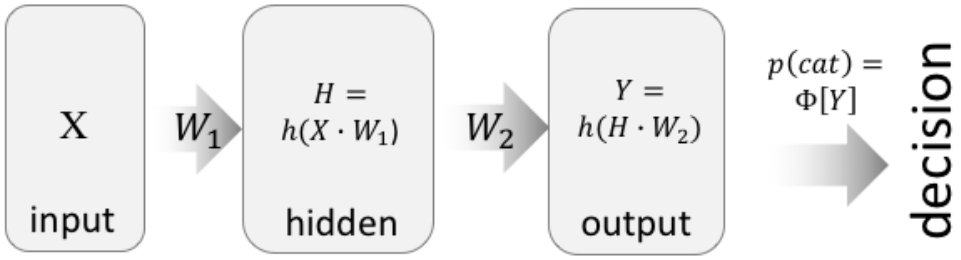


The perceptron can solve linear but not nonlinear classification tasks

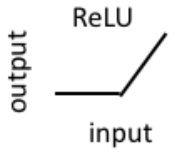
This simple problem halted progress in AI research for ~20 years, leading to the first "AI winter".

You can't draw a straight line that separates A from B!

What do we mean by a linear problem? In the graph above, I have plotted the possible values taken by a network with 2 inputs, x_1 and x_2. Let's imagine for simplicity that inputs are either +1 or -1. A problem is linear if there is a straight line that can be drawn through the space defined by values of x_1 and x_2 that cleanly separates the inputs into their respective categories. In the example given, the inputs $x_1 = 1, x_2 = 1$ and $x_1 = -1, x_2 = -1$ belong to one category, whereas the inputs $x_1 = -1, x_2 = 1$ and $x_1 = 1, x_2 = -1$ belong to another. As can be seen, there is no single straight line that can be drawn in these 2 dimensions that separates the inputs into 2 categories. The problem shown is known as the XOR problem. The failure of perceptrons (and sigmoid neurons) to solve this problem vastly dampened initial enthusiasm about AI research and let to the first "AI Winter", a wholesale withdrawal of funding from AI research and substantial slowdown in the field's progress.



Information is now transformed over "hidden" layers [here, 1] with a nonlinear activation function $h[\cdot]$ at each stage. The most common activation function is a rectified linear unit (ReLU)



Negative input set to zero
Positive input unchanged
Encourages sparse inputs

Luckily, however, there is a solution to this problem, but it took AI researchers a while to work it out. In hindsight, it's simple: nonlinear problems require nonlinear computations. One way to make your network learn nonlinear solutions is introduce an additional "hidden" layer to the network, in which the activations are a nonlinear function of the inputs. This is achieved by computing a hidden activation $H = h(X \cdot W)$ where $h(\cdot)$ denotes a nonlinear transform of some sort. A simple and effective nonlinear transformation is to set all negative activations to zero; this is knowns as a rectified linear unit or ReLU. This encourages the network to form sparse activations, which as we shall see later in the course, tend to help increase network capacity. The output Y is now computed not directly from X, but by passing H through another set of weights. This class of neural network is typically known as a "multilayer perceptron", even though technically it involves the stacking of sigmoid neurons[30].

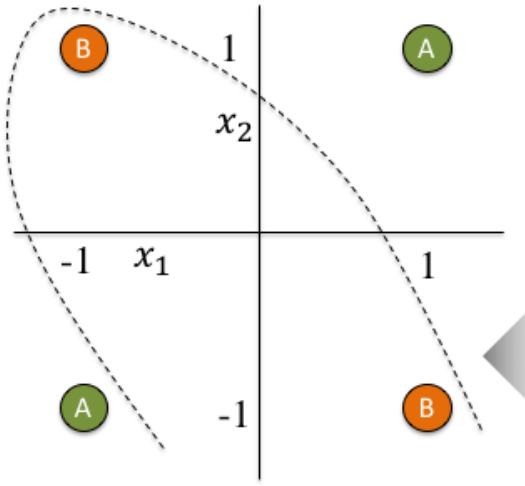
For our purposes, the critical intuition is that depth (in conjunction with nonlinear transduction) helps the network solve complex classification problems. This gives us an insight into why the primate visual system contains multiple stacked processing stages (V1, V2, V4, IT etc) and why neuronal firing rates tend to be a nonlinear function of their inputs.

[30] Have a look at these books, they are extremely helpful: *Neural networks and deep learning*. Nielsen (2015). *Deep Learning*. Goodfellow, Bengio & Courville (2016)

$$X^A = \begin{bmatrix} 1, -1 \\ 1, -1 \end{bmatrix} \quad X^B = \begin{bmatrix} -1, 1 \\ 1, -1 \end{bmatrix}$$

Rows are different features (e.g. x_1) and columns are different exemplars

By including multiple layers with nonlinearities, the perceptron can be adapted to solve a wide range of problems



one of many curved lines that can separate A from B

How can the MLP solve the XOR problem? By including multiple layers with nonlinearities, the network can learn a "curved" (nonlinear i.e. "not [straight] line") boundary through the input space, as shown on the figure.

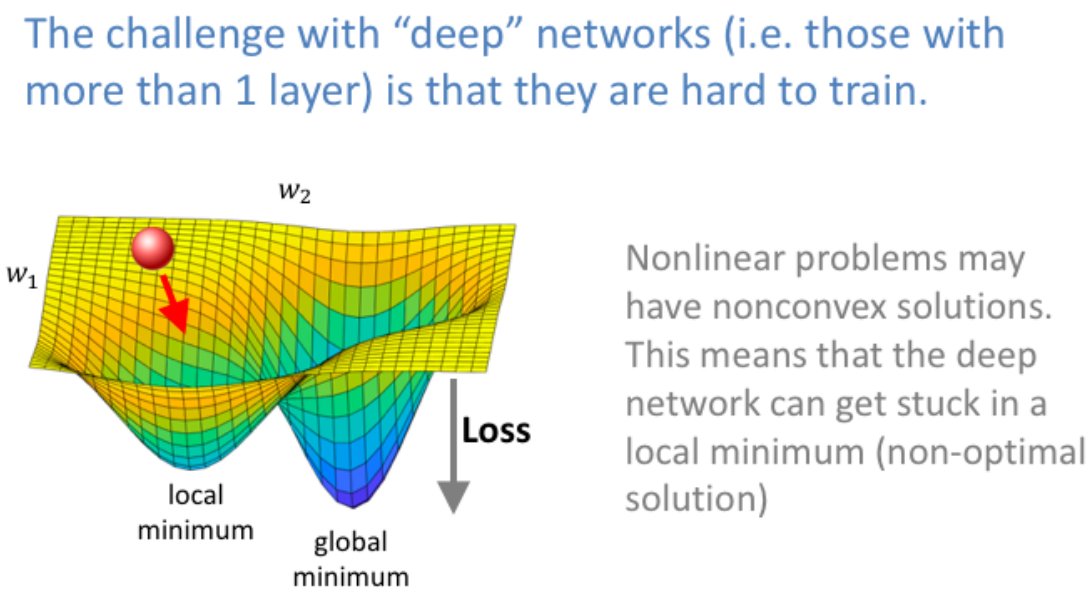
3.5. Challenges: optimisation, generalisation, and overfitting.

The connectionist movement began in the late 1970s with the discovery of a way to train multilayer ("deep") neural networks

The solution, known as **backpropagation**, is quite mathematically involved, but allows gradients to be computed in (backwards) succession from the output layer to the input layer, according to the "chain rule"

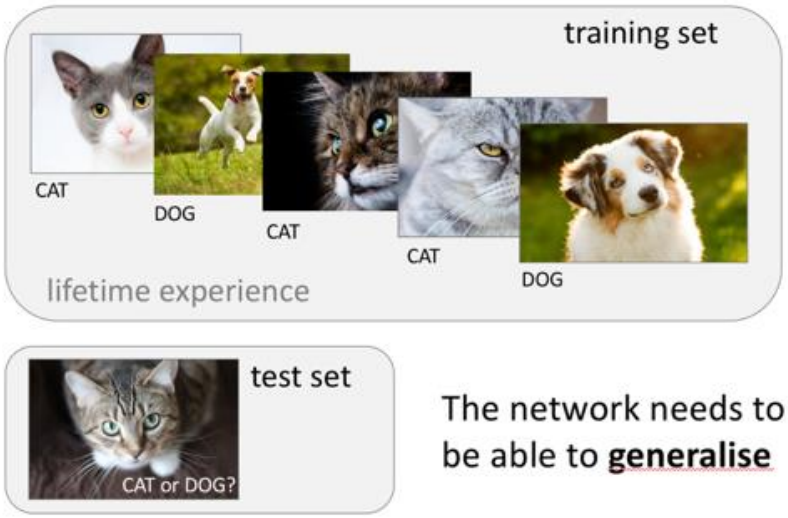
This class of network wasn't really built until the 1970s. So why did it take so long to work this out? Surely it should have been obvious that nonlinear problems require nonlinear solutions? Well, the answer is that it was obvious, but the tricky part is knowing how to *train* a network with multiple layers. The breakthrough came with the discovery of *backpropagation*, which is

a method that allows the gradients to be computed successively, from the output layer back to the input layer, via a principle known as the chain rule. A detailed explanation of backpropagation is beyond the scope of this course, but there are numerous online explanations for those who are interested[31].



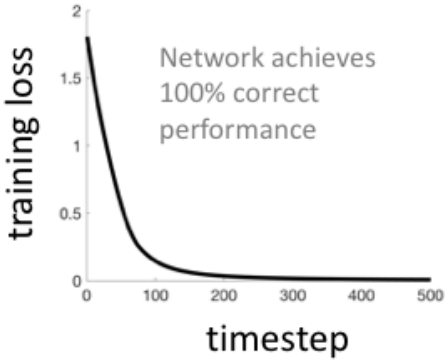
However, although backpropagation is a powerful technique, learning is hard in deep networks. This is because nonlinear problems often have *nonconvex* solutions. That means that in addition to the global minimum, there may be one or more *local* minima in the loss landscape. If the weights fall into a local minimum, gradient descent may not be able to pull them out, because a move in any direction will increase, rather than decrease the loss. This is a ubiquitous problem in deep learning.

[31] For example, https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/



Even if you find the global solution for your training set, that may not be what you want

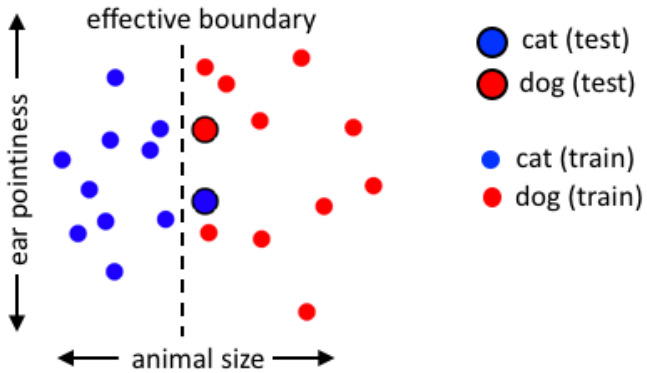
We began this lecture by pointing out the infinite variety of sensory experience. The key advantage of parametric models (such as a neural network) over nonparametric models (such as the lookup table in standard RL models) is that they can *generalise*. Of course, what we want is not just for the network to be able to classify the inputs that we have trained it on. We know what the labels are for those inputs – we had to, in order to be able to provide the supervision signals! What we want is to train the network on a dataset X for which we know the right answer (or "ground truth") but then test it on a new set of inputs for which we want to know the answer. This is of course similar to what happens during human development: we are taught what objects are ("that is a cat") and then we can identify new cats without further training.



Imagine you train a network on this training set, and it converges to loss ~0

Now you test on this left out set, but it is at chance! Why….?

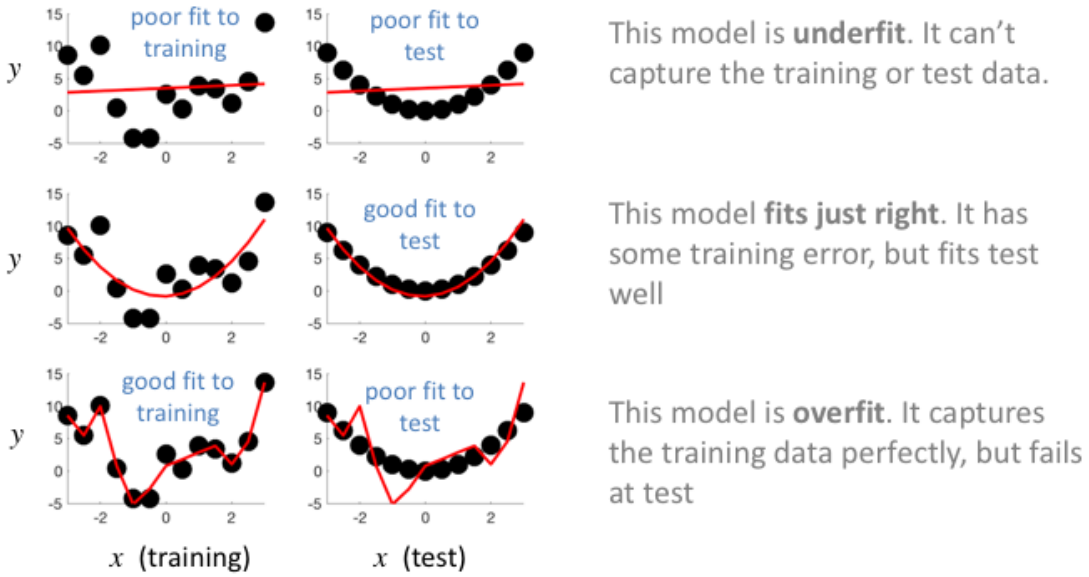
However, finding the right training set can be really tricky! Let's take an example. Imagine that the network is trained to classify the set of images shown on the upper left of the figure above as "cat" or "dog". The loss is plotted on the upper right: it converges nicely to zero, showing that the network has learned. Now, we test it on some unseen examples, like the ones on the bottom left, but it is completely unable to distinguish them. What has happened?



The network has learned to classify according to a single variable: number of black pixels.

It hasn't learned anything about cats and dogs at all!

Well, there are lots of things that could have gone wrong, but here's one example. If you notice, the training set varies on a number of features that might distinguish cats and dog. For examples, you recognise the cats by their distinctive cat shape. But in the training set we have chosen, they also differ on something much simpler – size! The dogs are all larger. The figure on the top right plots two dimension that the network could learn: size and ear pointiness. You can see that there is a boundary that perfectly segregates cats from dogs – all the network needs to do is learn to count the number of black pixels in the image (animal size). So, it learns to do this, and if our test set involves exemplars that do not differ in size (larger dots in the figure), it won't be able to tell them apart. The network hasn't learned anything specific to cats and dogs at all – it's just learned to be a size classifier.



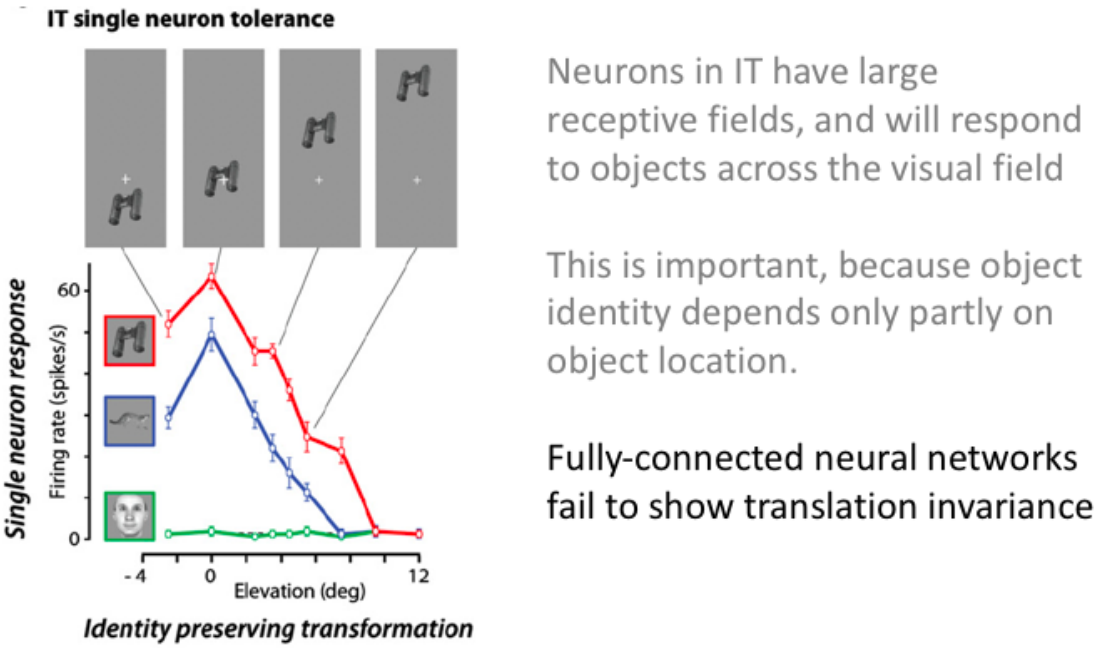
Simple example of overfitting with ground truth $y = x^2 + \mathcal{N}(0, \sigma)$

This is one example of how a network can overfit to the *training* dataset. This means that it learns something very specific which is true in the training set but not true in general. In general, the probability that this will happen depends on network size. If you make the network too big, it will overfit your data; too small, and it will underfit, because it has insufficient capacity.

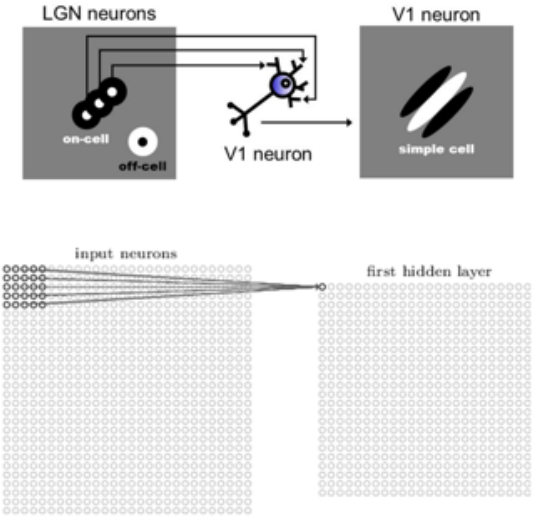
To illustrate this point, In the figure above, I show some (noisy) training data generated by the noisy quadratic function $y = x^2 + noise$. I then show the fit of various polynomial models that are trained on these noisy data but asked to generalise to the ground truth (non-noisy test data). A model with just 2 parameters (i.e. a first-order polynomial) underfits the training data, i.e. it doesn't have the expressive power to capture the quadratic form of the function. A model with 11 parameters does a fantastic job – capturing every small wiggle in the data. However, when we look at the (held out, i.e. unseen) test data, it fits badly, because it has learned to fit to the noise in the data, rather than to the underlying signal, which is the part of the data that is consistent between training and test. A second order polynomial, however, fits the data just right.

4. Structuring information in space and time

4.1 Convolutional neural networks and translation invariance



Feedforward networks can be powerful tools for data processing, but they don't generally do very well at classifying natural images. This is because of the multiple sources of variation that occur in natural images. In an image of a cat, that cat might be big or small; or on the left or the right of the image. One of the salient hallmarks of neurons in the more anterior parts of the primate ventral stream is that they display *invariance* to size, rotation and scale of the object(s) that they are selective for. An example is shown above. This cell in macaque IT responds vigorously to a pair of binoculars when presented at fixation, and although the response drops off somewhat as the object is translated vertically, it still exhibits a strong response. This neuron has (a degree of) translation invariance for its preferred feature. Feedforward neural networks essentially learn to classify images based on individual pixel values and their interactions; the network would have to see binoculars in every possible location before it was able to satistfactorily classify them in a position-invariant fashion. This makes them very *sample-inefficient*.



In the mammalian visual system, processing is convergent and cells have spatially-specific receptive fields

CNNs mimic this by assuming **local filters** - weights from a small group of input units to a single neuron in the hidden layer

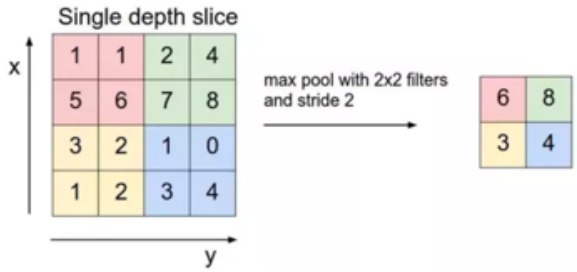
Each local filter (with fixed weights) is convolved with multiple overlapping positions on the input map, so that the responds at hidden unit {x,y} is now n deep, where n is the number of positions

CNNs combine local filtering and global weight sharing

Krizhevsky et al 2012

The real breakthrough in image recognition using neural networks came with the invention of the convolutional neural network (CNN)[32]. The CNN builds in an algorithmic feature that ensures translation invariance, and it does so by copying a salient feature of biological visual systems – that rather than each neuron receiving data from every single input location (e.g. image pixel), network units have spatially selective receptive fields, i.e. they learn a (local) filter that is specific to a location in space. So for example, in a 100 x 100 pixel image, a unit in the first hidden layer might receive inputs only from the first 5 x 5 square of pixels in the top left hand corner. Critically however, each filter is "shared" across multiple regions of space, with the resulting activations stacked along a separate dimension, as if each unit were not a single neuron, but a bank of neurons with distinct RF locations but the same tuning properties. In this way, the network is able to "share" information learned at one location (i.e. that pointy ears in the top left corner predicts "cat") with other locations, so that the network can learn to identify a cat by the presence of pointy ears in any location on a subsequent image. This type of spatially selective, convergent processing across local filters that tile the input space is key algorithmic feature of the visual system in mammals.

[32] The classic paper for convnets is this one: http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf



Pooling downsamples the image to reduce dimensionality

7	6	2
7	1	4
6	7	3

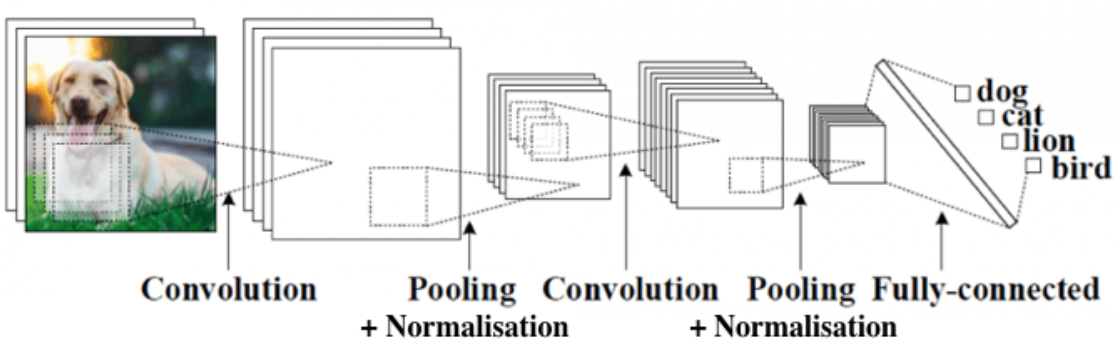
Each cell is normalized by its neighbours to accentuate local differences, producing an effect similar to lateral inhibition

$$b_i = \frac{a_i}{\left(k + \alpha \cdot \sum a_j^2\right)^{\beta}}$$

See Carandini & Heeger 2012

CNNs incorporate other features that are characteristic of biological vision. Firstly, at each stage the image is downsampled via a pooling operation. This ensures that the dimensionality of the inputs is reduced at each layer, much like we saw above in the primate visual system. This compression increases the efficiency of image representation. Secondly, CNNs often use gain normalisation[33]. Normalisation is a canonical feature of computation in neural circuits, including the visual sytems, and works by dividing (or normalising) activations in a particular cell by the local average activation, thereby accentuating differences. It is well known that visual circuits employ similar principles, for example via lateral inhibition among simple cells in V1.

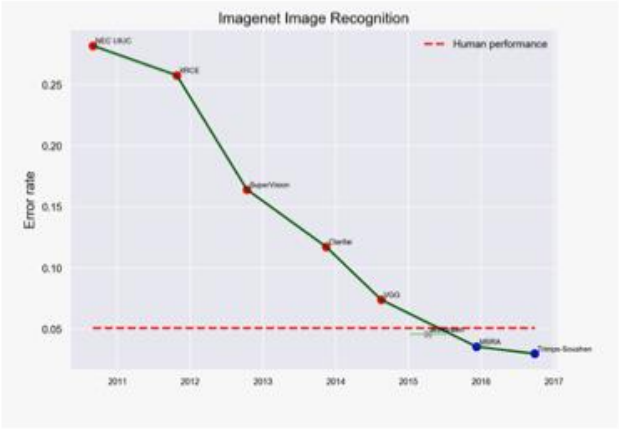
[33] Carandini, M., and Heeger, D.J. (2012). Normalization as a canonical neural computation. Nat Rev Neurosci *13*, 51-62.



The image is recursively filtered over successive layers to generate an "abstract" representation which is then decoded into an object category

Complex behavior emerges from simple operations!

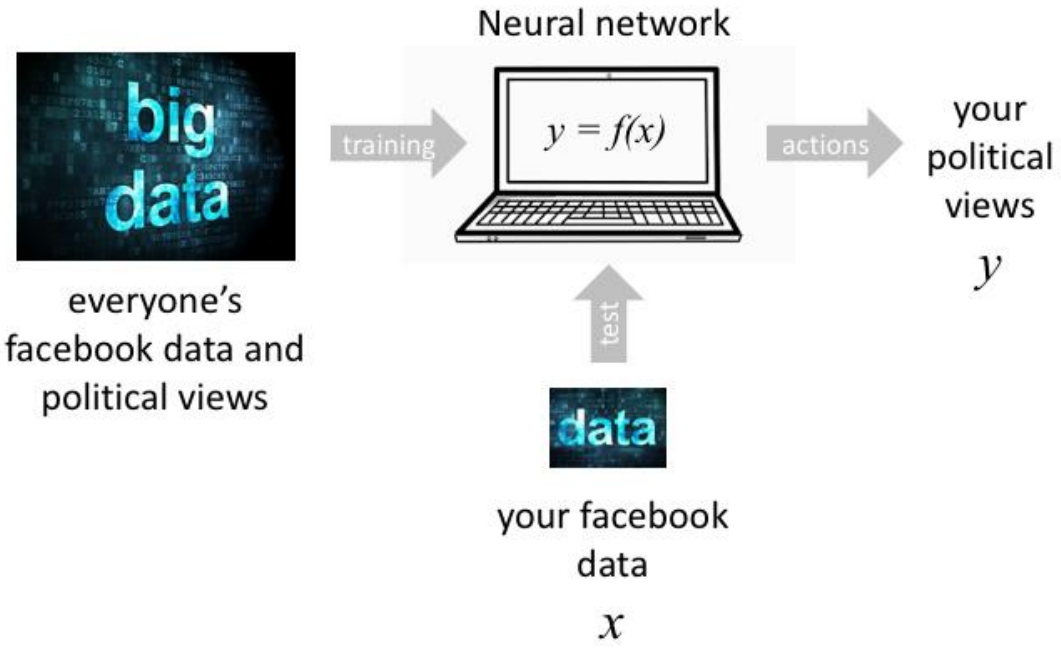
This slide shows a schematic of a simple CNN, including the various computational operations it involves: nonlinear transduction (as in the MLP), but also local filtering, weigh sharing, pooling, and normalisation. These convolutional layers are typically followed by a set of fully-connected layers, as in an MLP. After training with gradient descent (via backpropagation), these simple operations, combined and stacked successively into multiple layers, allow complex, high-dimensional image signals to be "disentangled" into a set of discrete class labels. Each layer of the CNN is a repeating motif with similar form; just like in neocortex, a simple algorithm (implemented in the canonical microcircuit) is repeated at each processing stage. Complex behaviour emerges from a succession of simple operations.



CNNs now outperform humans on the imagenet challenge, which involves classifying millions of previously unseen (test) images

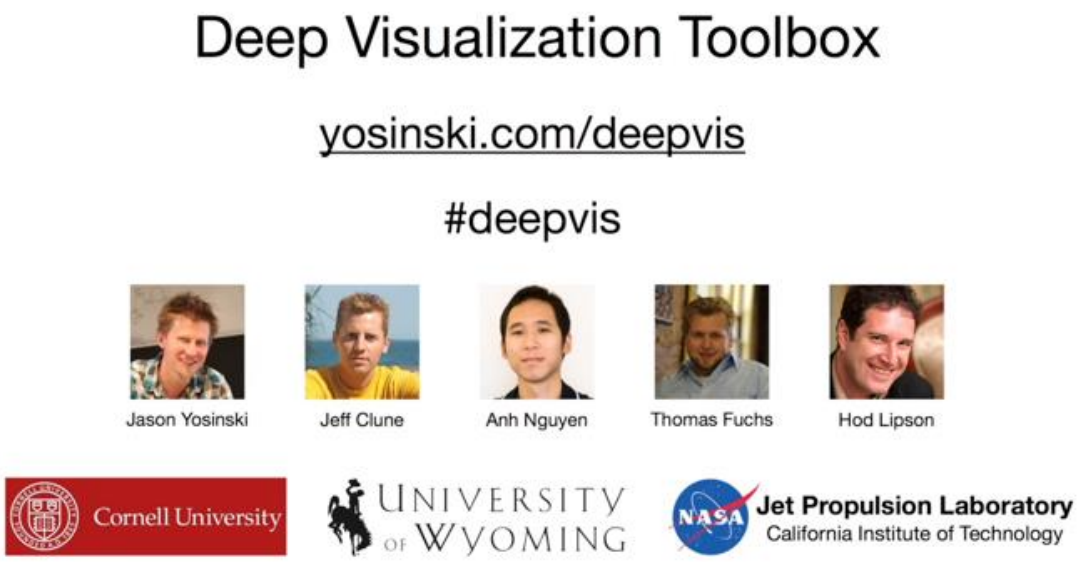
So what can CNNs actually do? Well, they are now very, very good at image recognition. A standard annual challenge, known as Imagenet[34], requires researchers to build a neural network that can classify a held out set from 1.2M labelled images of natural scenes, into the correct category (from 1000 possibilities). Adult humans typically display ~5% error on this task (they are not perfect because some of the images are unclear, and some of the categories are quite obscure). On the right, I have plotted the performance of the winning network over recent years. The most substantial drop in error came in 2012, with the introduction of CNNs. State of the art networks, built on the principles we have outlined here, now perform at < 3% error, i.e. they show "superhuman" image classification capacity.

[34] http://www.image-net.org/

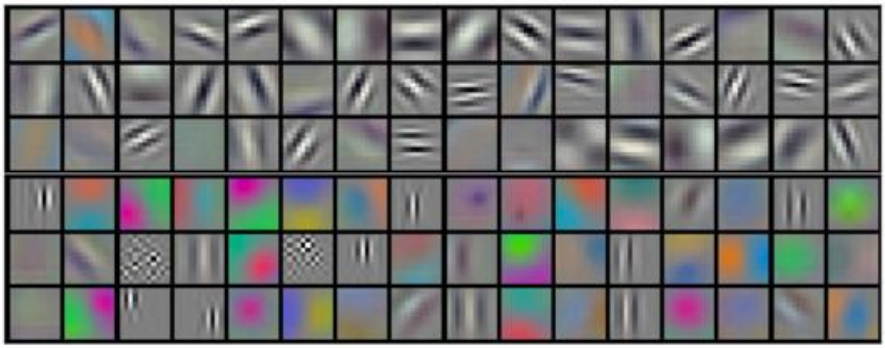


The power of neural networks, and in particular CNNs, is being felt everywhere, and driving a revolution in data processing (and the birth of the field of "data science". The same principle holds as for image recognition: if you have a large labelled dataset, you can train a neural network to make predictions about new, unlabelled data. The applications of this approach cross the domains of health, education, marketing and scientific research. Sometimes the uses can be controversial. For example, in 2018 it came to light that Cambridge Analytica, a data science company, had collected a large body of data from people's Facebook pages, and trained a neural network to predict their political preferences. This allowed the company to predict, from any new Facebook page, for whom they were likely to vote. This information was, understandably, of great interest to political campaigners, who were sold the predictions so that they could direct personalised political advertising towards Facebook users. This led to accusations that the company had unfairly biased the outcome of the 2016 US Presidential election and UK referendum on membership of the European Union.

4.2. Convnets and the ventral stream



After training, CNNs develop neural representations, just like the mature visual system in mammals. To what extent do they resemble the response properties found in biological systems? This video[35] shows some beautiful examples of the sorts of features that individual CNN units become sensitive to, including faces, textures and text.

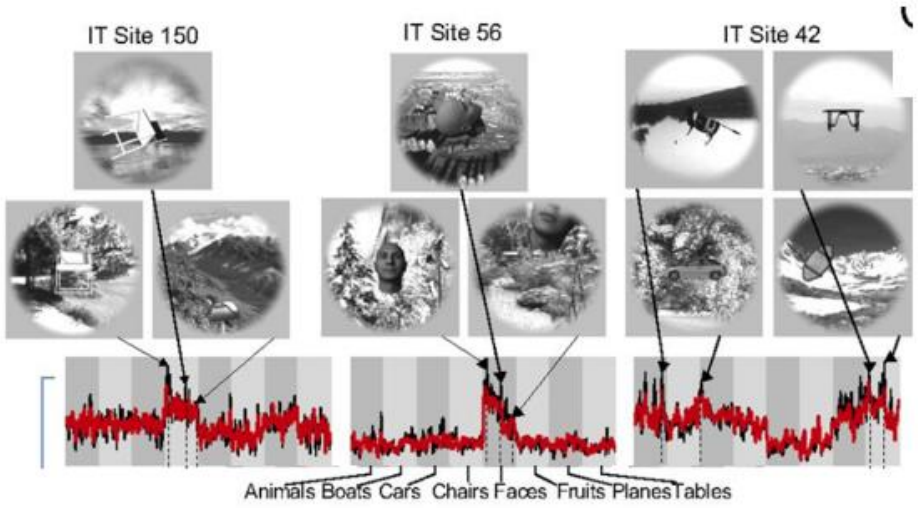


Filters learned by first layer of CNN resemble coding properties of V1 cells, i.e. Gabor filters with variable orientation and spatial frequency

We began by asking whether deep neural networks provide a plausible computational theory of how the representational properties of visual neurons emerge. This is still a matter of

[35] http://yosinski.com/deepvis

considerable contention. However, there is evidence to suggest that they do. One striking feature of CNNs is that after training, the units in their early layers form representational properties that resemble those in the early stages of biological visual systems, developing filters that are orientation and spatial-frequency selective or that display colour opponency (as in V1).

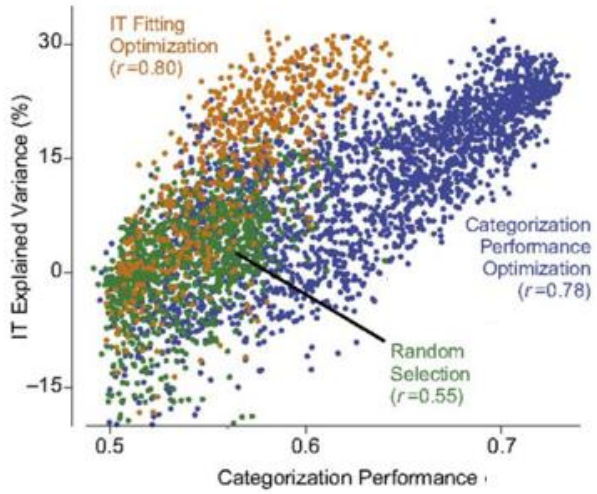


Strong overlap in responses of IT neurons and network units to different classes of stimuli

Yamins et al 2014

There is also evidence that the preponderances of neural selectivity for different image classes in the higher layers match those in primate IT. For example, the figure above shows a plot of the average network response to different classes of object (e.g. animals, cars, faces, tables) in red, overlaid on the selectivity of IT neurons recorded from different sites. The convergence is striking! [36]

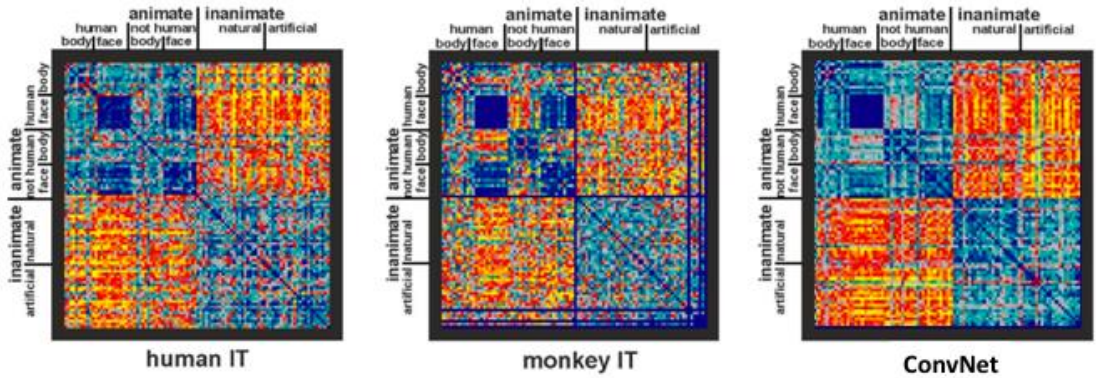
[36] Yamins, D.L., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., and DiCarlo, J.J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc Natl Acad Sci U S A *111*, 8619-8624.



Better-performing networks resemble IT more closely than poorer-performing ones (and just as well as networks optimized to resemble IT)

Yamins et al 2014

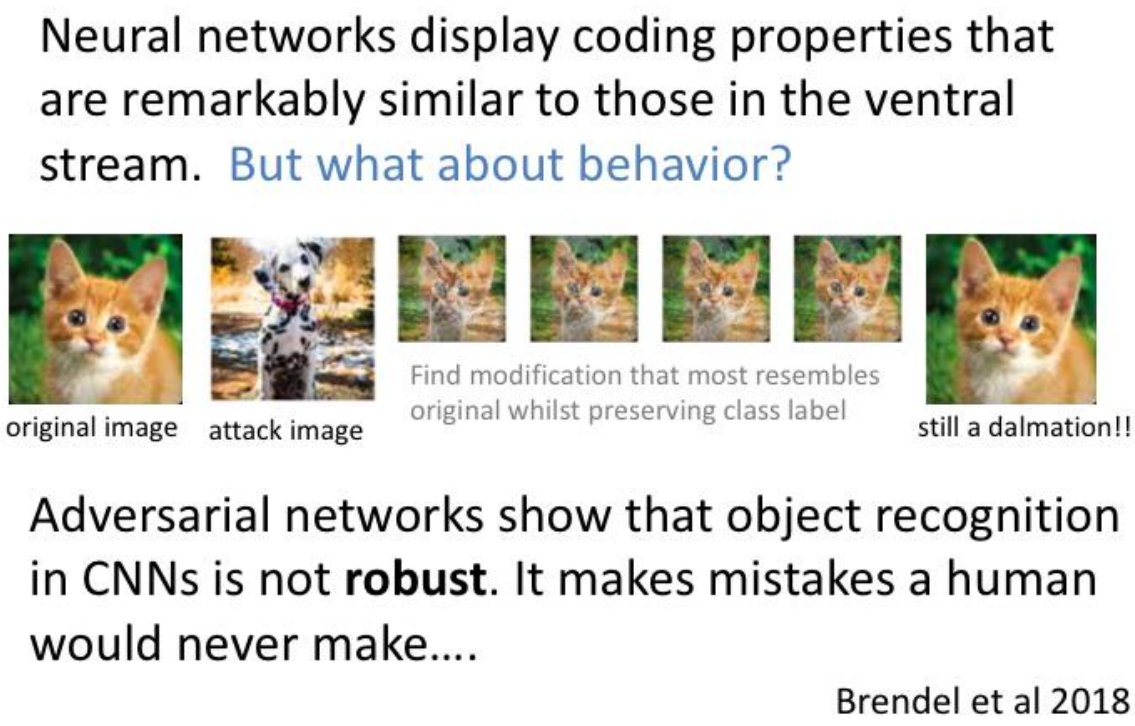
In fact, it's salient that this similarity in coding properties emerges even when the network is simply trained to perform well, rather than optimised to recreate the behaviour of biological cells. In fact, if the network is trained so that the loss reflects discrepancy with IT data, rather than object label classification error, then the variance in IT data explained is quite high, as would be expected. However, it's just as high (or higher) if the network is trained simply to perform well on the task, and the better it performs, the more the resulting activations resemble the responses of IT cells.



Representational similiarity of BOLD signals (in humans), firing rates (in monkeys) and unit activations (in CNNs) exhibits a consistent pattern for a range of naturalistic objects

Another, potentially more powerful means to visualise the coding similarity between deep neural networks and IT representations is to use a technique known as *representational similarity anlaysis* (RSA). RSA measures the pattern of neural responses (across multiple cells, or voxels in an fMRI experiment; or unit activations in a neural network) and computes the degree of similarity among patterns elicited under various experimental conditions. Thus, if your inputs are image classes, you can measure the pattern for cats, dogs, faces, cars and tables, etc and compute the $n \times n$ similarity matrix where n is the number of classes. On the figure above, I have visualised similarity matrices for a large number of object classes for human extrastriate visual cortex (from BOLD), cells in monkey IT, and the unit activations in an CNN. There is a fair degree of similarity between the CNN and the neural data (although it is mainly driven by a distinction between animate/inanimate objects and a preference for faces)[37].

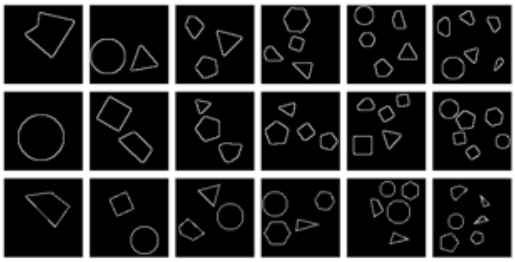
4.3. Limitations of deep networks



Are deep networks a plausible model of the primate ventral stream? Well, above we have seen evidence that they are. However, it is also important to point out some limitations of deep networks. One recent challenge has come from experiments which show just how startlingly easy it is to fool them. One approach, known as *adversarial* methods, trains a separate network

[37] Khaligh-Razavi, S.M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS Comput Biol *10*, e1003915, Kriegeskorte, N., Mur, M., and Bandettini, P. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. Front Syst Neurosci *2*, 4.

to try to find images that the client CNN will classify incorrectly. In the example above[38], the network has learned to classify (house) cats and dalmations. The adversarial network "reads the mind" of the CNN and tries to adjust one of the images incrementally until it maximally resembles an image of a different class, but without losing its class label. The adversarial network is able to find images that the CNN still thinks are a dalmation, but which are, to the human eye, definitely not a dalmation. In other words, CNNs learn to perform accurately at image recognition on average, but they make mistakes that a biological system would never make. CNNs do not learn robust policies for object recognition – they can classify, but appear not to *understand*, what objects are. We shall return to this issue in lecture 6, when we discuss unsupervised methods.



n \ m	1	2	3	4	5	6
1	0.687	0.313	0	0	0	0
2	0.026	0.390	0.583	0.001	0	0
3	0.002	0.006	0.021	0.896	0.075	0
4	0	0	0	0.014	0.492	0.494
5	0	0	0	0.001	0.043	0.956
6	0	0	0	0	0.012	0.988

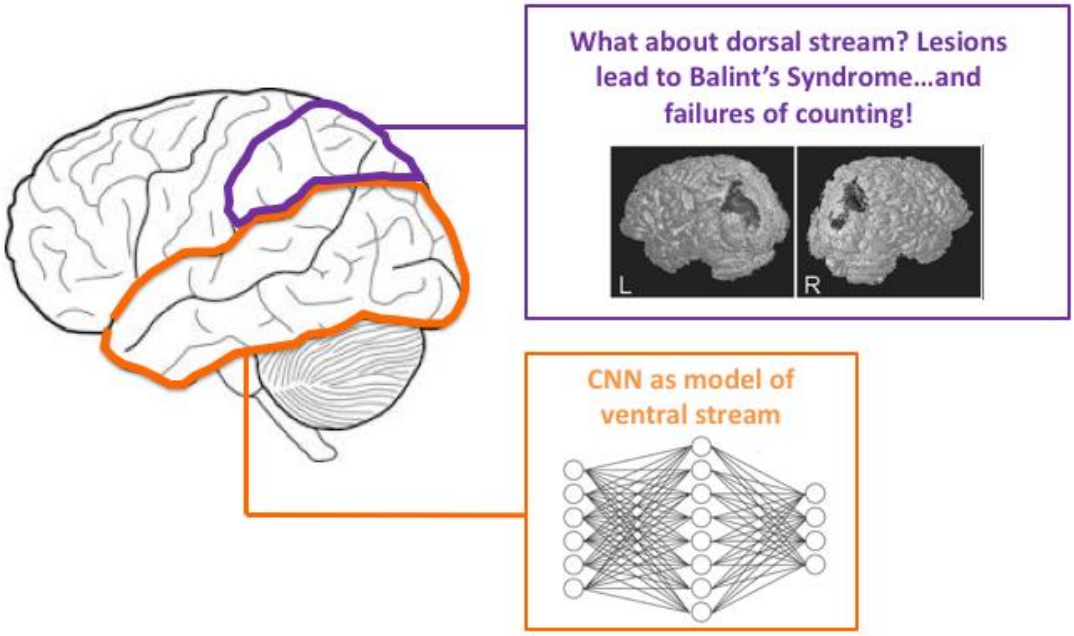
Should be a much simpler task than 1000-way object recognition…but CNNs fail badly

Why is this?

Wu et al 2018

There are other simple things that CNNs fail badly at. One is counting. The images above are much less complex than the natural scenes in the Imagenet competition. But a neural network that can learn to perform image classification with a high level of accuracy fails at the simple task of stating whether there are 1,2,3,4,5 or 6 shapes in the image[39]. Why is this?

[38] https://arxiv.org/abs/1712.04248
[39] https://arxiv.org/abs/1802.05160

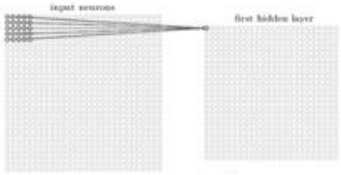


One possible answer is that deep neural networks may be a plausible model of the ventral stream, but object recognition in humans relies on both the dorsal and the ventral stream. In fact, patients with dorsal stream lesions – such as in Balint's syndrome – exhibit many of the same characteristic deficits of trained CNNs, in that they have difficulty counting, or performing matching or comparison judgments for novel stimuli[40]. Later in the course, we will discuss ways in which AI systems might be augmented such that they more closely resemble the full functioning of the primate visual system.

[40] Friedman-Hill S, Robertson LC, Treisman A. Parietal contributions to visual feature binding: Evidence from a patient with bilateral lesions. Science. 1995;269:853–855

4.4. Hierarchies of temporal integration in the brain

Sensory signals are spatially structured

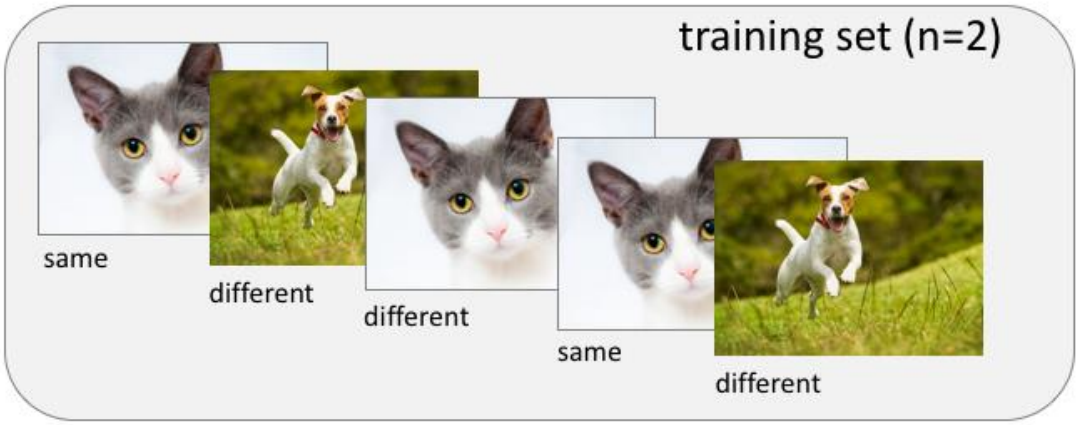


CNNs build in priors (or "inductive biases") that objects should exhibit consistent spatial structure

But they are also **temporally** structured.

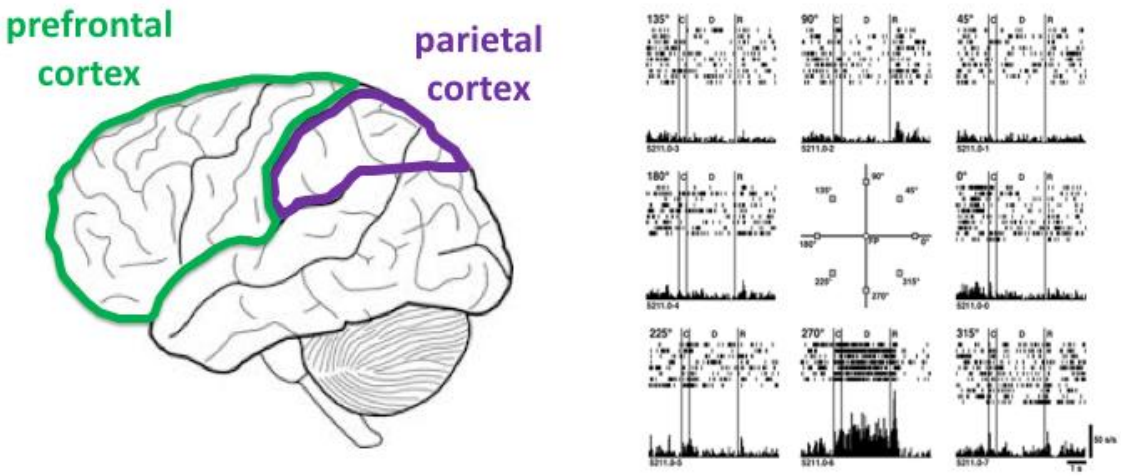
Feedforward neural networks cannot capitalize on this structure, because they have no activation memory

In the previous lecture, we encountered convolutional neural networks (CNNs). CNNs take advantage of the spatial structure of natural images: nearby pixels tend to be related in a fashion that predicts the object class. This is what allows the success of methods that filter locally, and then share filters across space. However, in the natural world, sensory signals are structured in time as well as space. In many domains it is necessary to process how sensory signals unfold in time to know how best to act, such as when judging the speed and direction of an oncoming vehicle, or when trying to understand a sentence spoken in natural language.



A feedforward neural network would perform at chance at this simple task, because it has no means to remember what occurred on the last timestep

The class of network we discussed in the previous lecture, known collectively as feedforward networks (a term that encompasses both MLPs and CNNs), has no way of processing information in time. This is because their activations are uniquely determined by the current input; there is no mechanism for information that was present in the network on the previous timestep $t-1$ to influence the current network state at time t. Thus, despite the power of CNNs for static image recognition, they would fail at other, very simple tasks. Imagine that the inputs are again a stream of images, but the task is now to output whether each image is the same as, or different from, the previous image. Standard feedforward networks would fail utterly to learn this task, even if there were only 2 images in the training set. This is because they have no activation memory.



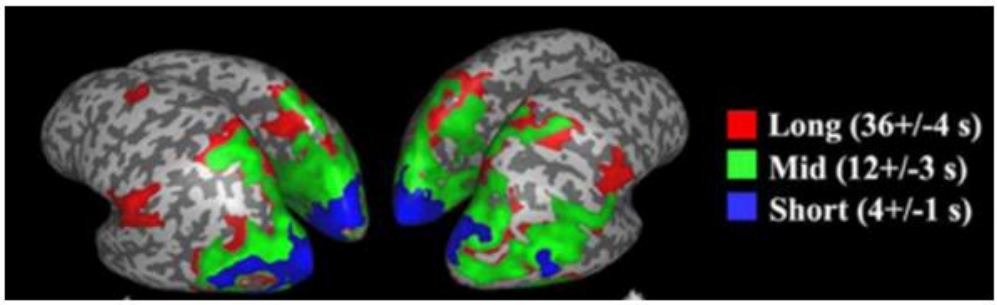
Neurons in lateral parietal and prefrontal cortex display persistent, content-specific activity in working memory tasks

e.g. Goldman-Rakic, Fuster

Of course, we know that this is not true of biological networks. In both rodents and primates, cells in higher association cortex, including the prefrontal and parietal cortices, exhibit persistent, content-specific activation, which is thought to form a substrate for short-term integration processes or working memory. Take, for example, the classic result on the slide above, which shows data recorded from a cell in the dorsolateral prefrontal cortex (DLPFC) of the macaque monkey during a task that involves the presentation of a spatial cue, its subsequent extinction. Following a delay period, a "go" signal prompts the monkey to make a saccade to the remembered location of the cue. During the delay period, cells in the DLPFC fire persistently in a spatially selective fashion. For example, the cell shown is tonically active when the cue was initially presented in the lower central portion of the screen, but not elsewhere[41]. Similar neurons are observed in the parietal cortex, and during object match-to-sample tasks, in the anterior portions of the temporal lobe discussed in the previous lecture.

[41] Goldman-Rakic PS. Cellular basis of working memory. Neuron. 1995 Mar;14(3):477-85.

Segments were then scrambled over different timescales, and the reproducibility (correlation in BOLD signals) was measured over repeats of the same segments



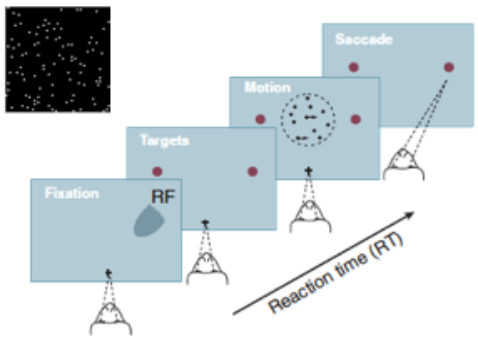
This revealed a hierarchy of temporal windows across the cortex from posterior to anterior

Hasson et al 2008

In fact, it is a principal of cortical organisation in primates that cells in more anterior regions are found to have longer temporal windows of integration. This is nicely demonstrated by this clever fMRI experiment from Hasson and colleagues[42], who showed participants silent (Chaplin) films, but jumbled the scenes over different timescales. For example, for some films the first and second half of the story might have been switched, whereas for others the overall story remained intact but short segments were mixed up within a short period. Using a technique that measured the extent to which BOLD signals correlated across the participant cohort, the researchers were able to identify brain regions that coded for information over short timescales (i.e. where brain signals were decorrelated by mixing scenes within short periods) and over long periods (where there was an invariance to such short switches, but disruption when the jumbling occurred over longer segments). They found that there was a hierarchy of timescales across the cortex, with early visual regions sensitive to short timescale information and more anterior regions to the longer episodes, presumably because of their more substantial involvement in processing the overall narrative of the film.

4.5. Temporal integration in perceptual decision-making

[42] Hasson, U., Yang, E., Vallines, I., Heeger, D.J., and Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. J Neurosci *28*, 2539-2550.



In a standard psychophysical task, observers judge the motion direction in a stream of randomly moving dots

solution is to compute the (log) likelihood ratio: $\log\left(\frac{p(x|R)}{p(x|L)}\right) \propto \left(\frac{p(R|x)}{p(L|x)}\right)$

From Bayes' rule, posterior is proportional to likelihood

Sequential probability ratio test (SPRT)

$$\log\left(\frac{p(x_{1\to n}|R)}{p(x_{1\to n}|L)}\right) = \log\left(\frac{p(x_1|R)}{p(x_1|L)}\right) + \log\left(\frac{p(x_2|R)}{p(x_2|L)}\right) \ldots + \log\left(\frac{p(x_n|R)}{p(x_n|L)}\right)$$

likelihood ratio after n samples | likelihood ratio for sample 1 | likelihood ratio for sample 2 | likelihood ratio for sample n

Wald & Wolfwitz, 1946; Bogacz 2006

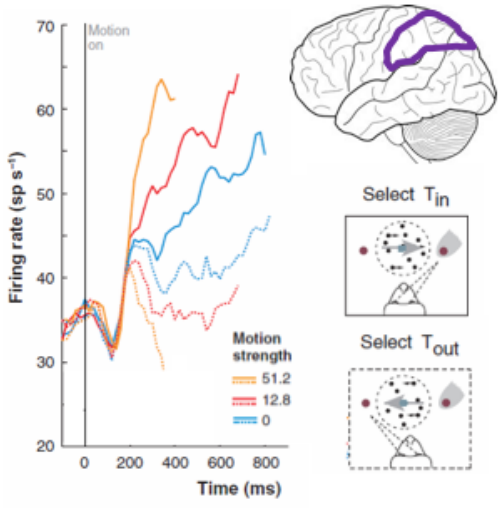
How does the brain integrate information over time, at the computational level? This question has been extensively addressed in the field of perceptual decision-making, where a long tradition has sought to define optimal principles for sensory integration and compare these to data from psychophysical experiments in which decisions are based on multiple samples of from an input stream[43]. In one canonical paradigm, a monkey or human views a psychophysical stimulus consisting of a cloud of moving dots. Some of the dots move randomly, but other move in a fixed direction, such as left or right. The observer's task is to report the net direction of motion. Because the motion direction signals are independent from frame to frame, any one pair of frames yields a very noisy estimate of the correct answer. However, a good estimate can be formed by sequentially sampling and integrating information across frames. This occurs because the noisy estimates will average out to zero over time, yielding a precision of the net direction estimate that grows with the number of samples taken (i.e. frames viewed). Indeed, as might be expected, human and monkeys perform more accurately under long than short viewing durations, as if they were successfully integrating sensory information across time.

Theoretical work dating back to the 1940s (and in fact, to Alan Turing's contribution to the British wartime effort during World War II)[44] has identified an optimal quantitative framework for understanding how information should be integrated to maximise certainty about the correct answer, given the fewest possible samples. A decision about the class label of a noisy sensory stimulus should ideally be based on the likelihood ratio, that is the relative probability

[43] Bogacz, R., Brown, E., Moehlis, J., Holmes, P., and Cohen, J.D. (2006). The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. Psychol Rev *113*, 700-765, Gold, J.I., and Shadlen, M.N. (2001). Neural computations that underlie decisions about sensory stimuli. Trends Cogn Sci *5*, 10-16, Wald, A., and Wolfowitz, J. (1949). Bayes Solutions of Sequential Decision Problems. Proc Natl Acad Sci U S A *35*, 99-102.

[44] Gold, J.I., and Shadlen, M.N. (2002). Banburismus and the brain: decoding the relationship between sensory stimuli, decisions, and reward. Neuron *36*, 299-308.

of observing the data given that is belongs to one category or the other. Where multiple samples of information are available, the respective likelihoods should be combined by multiplication; or equivalently, by summation of log likelihood (ratios). This approach, known as the sequential probability ratio test (SPRT), grandfathers most current models of sensory integration in biological brains, and was – incidentally – of great help in cracking the Enigma code during war.



During decisions about a dot motion stimulus, firing rates increase in proportion to the signal-to-noise ratio (motion coherence)

Neurons in the lateral parietal and prefrontal cortex act as integrators during perceptual choice tasks

$$V_i = V_{i-1} + \delta + \mathcal{N}(0, \sigma)$$

cumulative DV · drift rate · noise

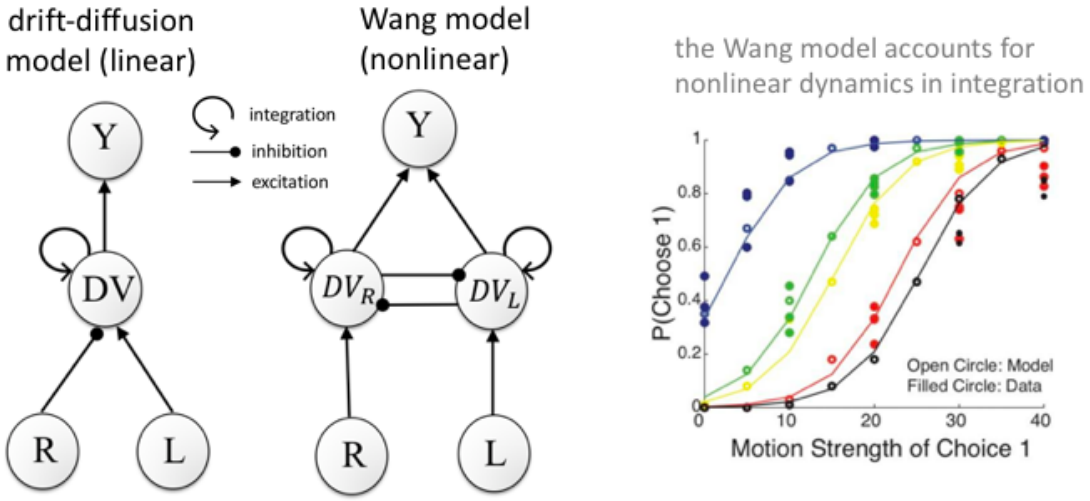
i here denotes the integration cycle within at trial

average firing rates and behaviour can be predicted by sequential sampling models such as the DDM, which approximates SPRT

Gold & Shadlen 2007; Hanks & Summerfield 2017

The theoretical framework furnished by the SPRT is popular because it provides the foundation for a theory that has enjoyed great success in jointly explaining psychophysical behaviour in paradigms such as the dot motion discrimination task, and the firing rates of neurons in the lateral parietal cortex[45]. Cells in lateral intraparietal (LIP; identified by their exhibition of persistent delay period activity) have spatially specific response fields, and researchers begin by identifying cells whose RF is congruent with one of the targets to which the monkey makes a saccade when responding (e.g. the location for signalling "right"). When the sensory signal is congruent with that target (e.g. the dots are moving right), these LIP cells show gradual increases in firing rate that scale positively with the signal-to-noise ratio of the stimulus, as if the momentary firing rate were signalling the degree of cumulative evidence for a given response. This pattern of neural activity can be explained if the cells were integrating (or adding up) the relative (log) evidence over time for one of the two responses, as predicted by the SPRT and/or related models that approximate it, such as the drift-diffusion model (DDM).

[45] Gold, J.I., and Shadlen, M.N. (2007). The neural basis of decision making. Annu Rev Neurosci *30*, 535-574, Hanks, T.D., and Summerfield, C. (2017). Perceptual Decision Making in Rodents, Monkeys, and Humans. Neuron *93*, 15-31.



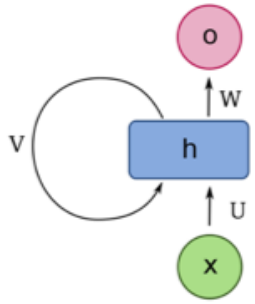
Elaborations of the sequential sampling framework include recurrent inhibition between integrators

Wang, 2012

Subsequently, this framework for understanding psychophysical judgments in settings that require sequential sampling and integration of sensory information has been elaborated with a wide family of competing models. Among the most biologically plausible of these is a model proposed by Wang and colleagues[46], which argues that sensory inputs (e.g. evidence for left or right) is fed forward to an intermediate layer, were both integration and mutual inhibition drive a nonlinear process that leads one of the responses to win a "race" to a decision threshold, through recurrent dynamics. This class of recurrent model has the benefit of biological plausibility, as well as the merit of doing a very good job of fitting both psychophysical and neural data.

4.6. Recurrent neural networks and the parietal cortex

[46] Wang, X.J. (2012). Neural dynamics and circuit mechanisms of decision-making. Curr Opin Neurobiol *22*, 1039-1046, Wong, K.F., and Wang, X.J. (2006). A recurrent network mechanism of time integration in perceptual decisions. J Neurosci *26*, 1314-1328.



$$H_t = h(U \cdot X_t + V \cdot H_{t-1})$$

current input — "memory" weight matrix — previous hidden unit activations

this allows the network to make sequential predictions on the basis of past information

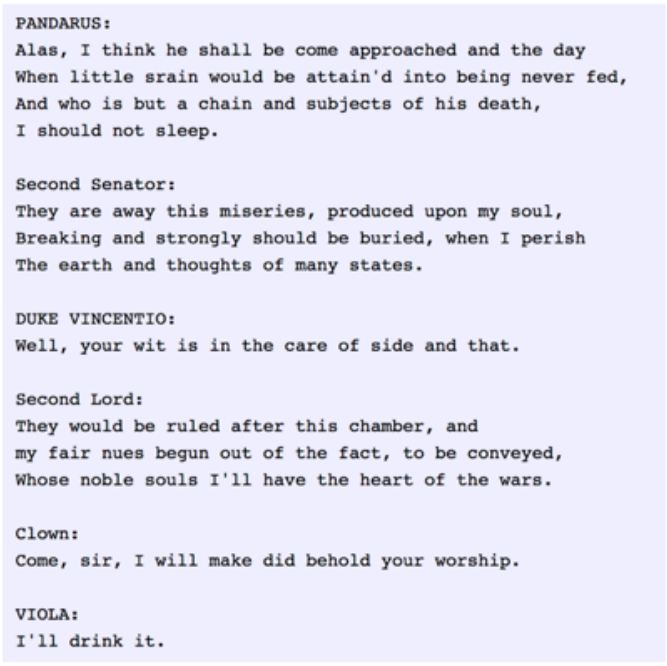
In a basic RNN, information flows forward through the network as in an MLP

However, the hidden units receive an additional input from themselves in the previous timestep

Starting in the 1990s, methods were developed that allowed neural networks to engage in a similar integration of information across time. The name given to this class of network is a "recurrent neural network" (RNN) and the kind of dynamics it is capable of generating – nonlinear, time-varying competitive interactions among inputs – is closely related to those displayed by biologically plausible models for perceptual decisions. However, the crucial difference is that recurrent neural networks have large numbers of freely trainable parameters, making them suitable for dealing with complex, real-world domains such as time-varying natural vision (e.g. video) and spoken or written language. This is unlike the parameters of models under the sequential sampling framework, which are typically hardcoded by the researcher.

The simplest recurrent neural networks extend the architecture of the multilayer perceptron (MLP) in one very simple way[47]. The inputs at time t in the hidden layer H_t, rather than being determined exclusively by the weights from the inputs layer (here, denoted U), are also driven by the previous state of the hidden layer H_{t-1} passed through a separate set of weights V. This simple addition allows a recurrent network to dynamically learn to maintain some information (and lose other information) between time steps, permitting decisions about data streams that are structured in time. Of note, the network can be trained to provide a sequence of outputs, making it the tool of choice for researchers interested in domains that required temporally structured motor plans, such as machine translation. Where appropriate, RNNs can be combined with convolutions and/or other successful algorithmic features of feedforward neural networks.

[47] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/



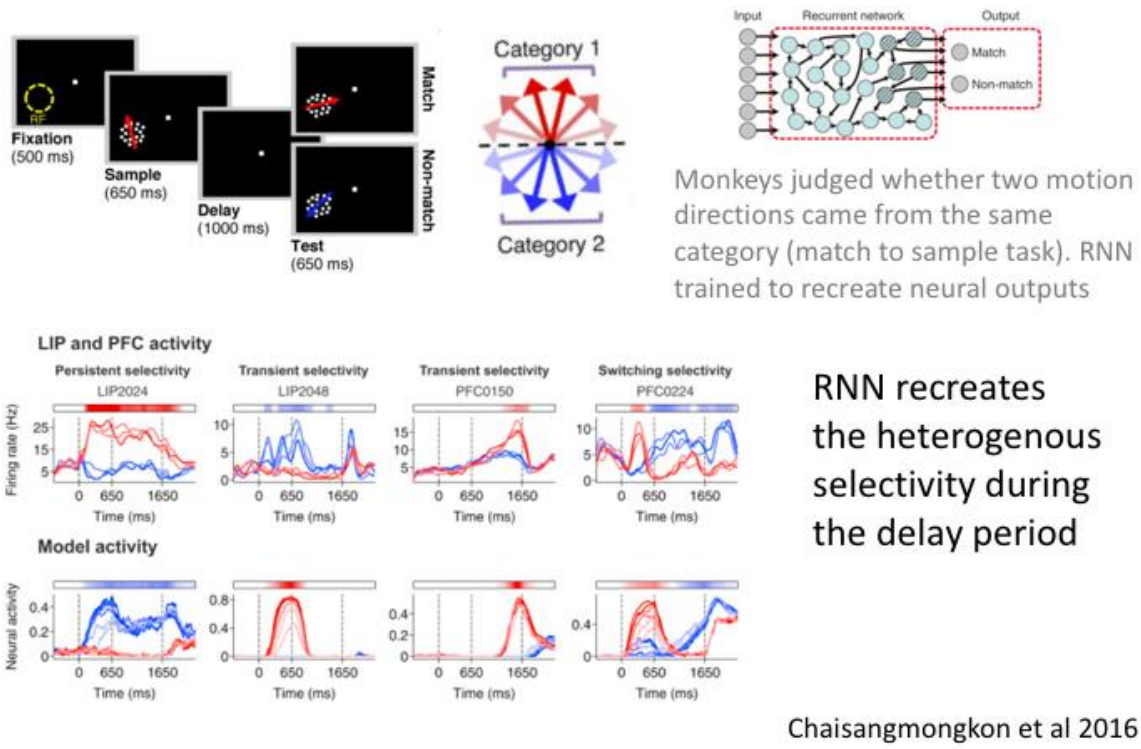
```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

And here is an RNN trained on the complete works of Shakespeare]

For obvious reasons, RNNs have widespread applications in machine translation

RNNs are powerful, as shown in the example above. Here, a recurrent network was trained to predict segments of text from the complete works of Shakespeare. A cursory glance at the output it produces might fool you into thinking that it has really learned to write a new Shakespeare play – it looks remarkably realistic! (However, on closer inspection, the text produced is largely nonsense[48]. The network has learned the vocabulary and syntax typical of Shakespeare's 37 plays – but not the meaning).

[48] Lots of nice examples here http://karpathy.github.io/2015/05/21/rnn-effectiveness/

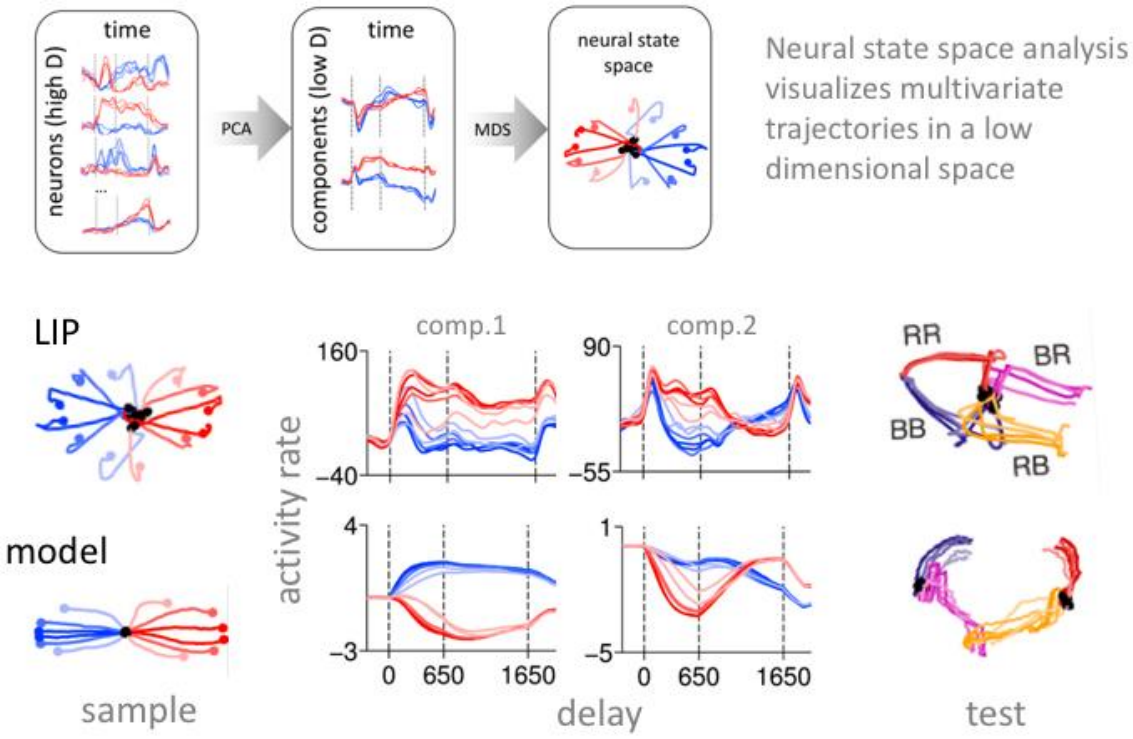


Over the last few years, researchers in neuroscience have come to use RNNs as candidate explanatory models for the coding properties of cells in the parietal and prefrontal cortices, just as we saw in the previous lecture that CNNs have been proposed as models of the ventral stream. Data from one exemplary study is shown above[49]. Here, the authors again used a dot motion stimulus in macaques, but rather than simply discriminating whether the dots tended left or right, monkeys were presented with 2 bursts of motion energy separated by a delay and asked to respond whether the two motion directions belonged to a same or different category. Category was defined by a line that cleaved the 360° space of motion directions, such that (for example) those directions that lay above the horizontal meridian (broadly, dots moving "up" – red arrows in the upper central panel) formed one category, and all others (blue; "down") formed another.

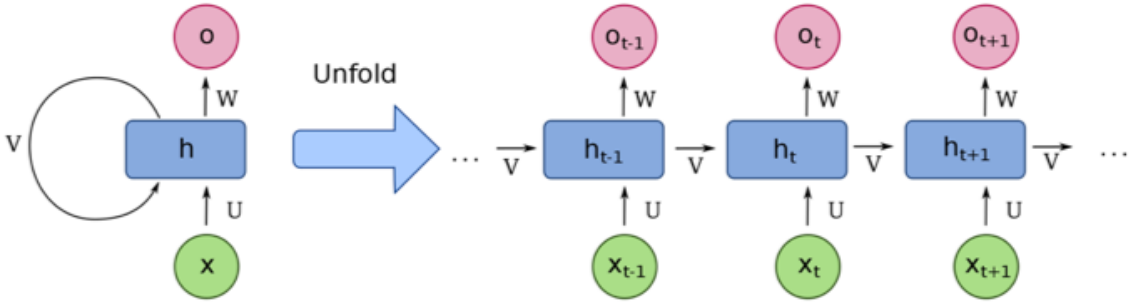
A key feature of neural data recorded in this task – and in many other psychophysical and working memory tasks – is the high degree of *mixed selectivity* exhibited by neurons in the parietal and prefrontal cortices. In other words, rather than simply responding with content-specific persistent activity – as reported in early working memory studies – neurons exhibit extremely varied coding properties and are variously sensitive to a range of task variables with a confusing array of time-varying response profiles. Some example cells from LIP and the DLPFC are shown in the figure above. The authors then trained an RNN to perform the task; the recurrent dynamics of the RNN allowed the network to maintain information across the delay period and make accurate match-to-sample judgments just like the monkeys did. Strikingly, "recordings" from the hidden units of the RNN model display an extremely similar

[49] Chaisangmongkon, W., Swaminathan, S.K., Freedman, D.J., and Wang, X.J. (2017). Computing by Robust Transience: How the Fronto-Parietal Network Performs Sequential, Category-Based Decisions. Neuron *93*, 1504-1517 e1504. See also this paper, which focusses on FEF rather than parietal cortex: Mante, V., Sussillo, D., Shenoy, K.V., and Newsome, W.T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature *503*, 78-84.

heterogeneity, even though the network was not trained in a way that guaranteed the emergence of mixed selectivity.



Building on these findings, in this study the authors took the analysis a stage further, using multivariate methods to define the "neural state space trajectory" of both the LIP/PFC neurons and the RNN units. The state space analysis takes the neurons x time matrix of neural activity, and compresses it using a dimensionality reduction technique similar to principal components analysis (PCA) to yield a (smaller) components x time matrix. In LIP, the first principal component extracted from the overall neuronal activity during the delay period encoded the (signed) level of disparity between the motion direction and the boundary, as shown in the left hand of the centre/lower panel (dark red and dark blue lines are furthest from boundary in each category; light red/blue are close to boundary). The second components reflected the temporal derivative of this activity, i.e. the extent to which it onset sooner or later. The same analysis on the RNN mimicked both of these components (inverted in these plots because PCA is rotation invariant). Subsequently, the neural "state space" is plotted as the values of the first principal component against another across time; this visualises a low-dimensional manifold on which the neural activity evolves at various points during the trial. As can be seen, both following presentation of the sample, i.e. the first motion stimulus (where neural activity again segregated according to distance to boundary) and following onset of the probe, i.e. the second motion stimulus (where the neural activity segregated according to whether the trial was a match or a mismatch), the RNN displayed highly similar dynamics. The authors present these results as evidence that the RNN offers a plausible computational account of the neural coding properties and their time-varying dynamics during this match-to-category task.

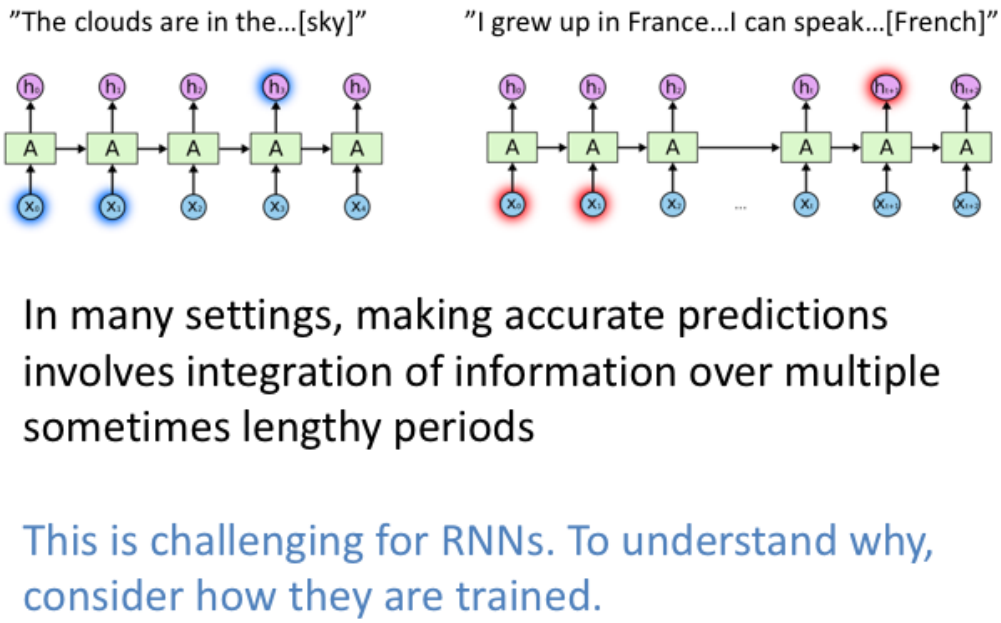


To train the network, the gradients need to be computed with respect to the past information. So the network needs to be "unrolled" in time, updates applied, and then "rolled back up".

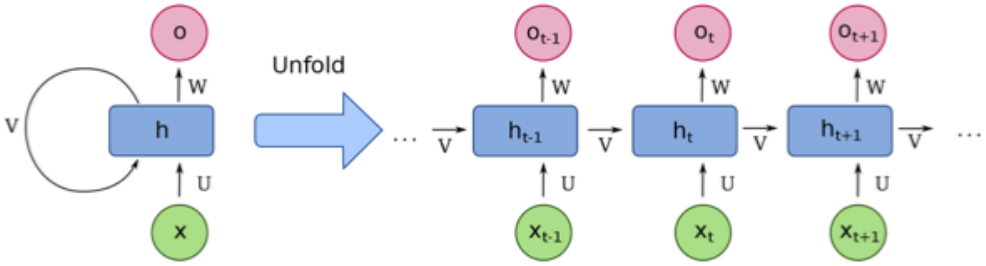
This leads to a number of problems

RNNs sound great, don't they?? They are powerful tools that look like plausible models of higher cortical activity. Well, RNNs are great, but like with many tools in deep learning, the great challenge is how to train them. This problem is particularly acute in RNNs, because they require a special training method, known as *backpropagation through time* (BPTT) which is both technically limited and biologically rather implausible. To understand how to train an RNN, it is worth revisiting the optimisation methods for feedforward neural networks. Recall that the MLP is trained with backpropagation, that is, the successive computation of the gradients for each computational stage of the network, from output through hidden back to input. The trouble with an RNN is that the activity states in both hidden (and consequently) output layers depend not only on their inputs on the current timestep, but on all the successive inputs they have received on previous timesteps! To accurately compute the derivatives and adjust the weights, thus, the network has to be "unfolded" so that each of its past states can contribute to the computation of the gradients[50].

[50] http://colah.github.io/posts/2015-08-Understanding-LSTMs/



The problem with this approach is that the network needs to retain a full memory of its past activity states over some reasonable time window. This is clearly both computationally cumbersome and biologically implausible. In particular, we might often want to make predictions about events that may be contingent on inputs received far back in the past. Consider two examples from natural language. Imagine I am trying to predict the last word in the sentence "The clouds are in the…". Here, I don't need to look too far back to find the relevant clue; I can integrate over a relatively short time window, and so BPTT is feasible. Imagine however, a passage of text that begins "I grew up in France…" followed by a set of other information, and the missing word follows the subsequent phrase "I can speak.." Here, the network might need a lengthy integration window – spanning perhaps an entire paragraph – to make a reasonable prediction. BPTT is poorly adapted for this sort of situation. Now, you might begin to see why that RNN-generated Shakespeare looked superficially plausible but lacked meaning – because the RNN doesn't have any sense of the narrative structure of the story, in part because it's temporal integration window is limited by the insufficiencies of BPTT. Of note, however, this might be less of a problem when considering relatively rapid discrimination or working memory judgments in the sensorimotor domain (e.g. made over just a second or two).



Problem 1: BPTT is computationally expensive. The size of the gradients grows with each additional timestep.

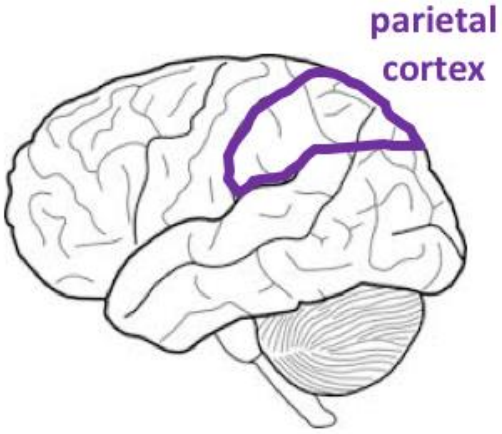
Problem 2: BPTT can lead to **vanishing gradients**, that occur when lots of small numbers are multiplied together via the chain rule.

Problem 3: BPTT over long timescales has limited biological plausibility!

In addition to the computational expense and limited of biological plausibility, RNNs are prone to another problem – that of *vanishing gradients*. In fact, this problem is not unique to RNNs, but occurs whenever a network is being trained using backpropagation through many separate computational stages (such as a very deep feedforward network). Vanishing gradients occur when lots of very small derivatives are multiplied together, potentially yielding a infinitesimally small update signal. All in all, the problem that RNNs suffer from can be cast in general terms: when dealing with time-varying information, it's hard to assign credit for outcomes that may occur several steps after the input that provoked them. It's rather like working out who infected you when you become ill with a virus that has a long incubation period – it could have been almost anyone! This problem, known as temporal credit assignment, is a major challenge for biological and artificial systems alike.

5. Computation and memory systems

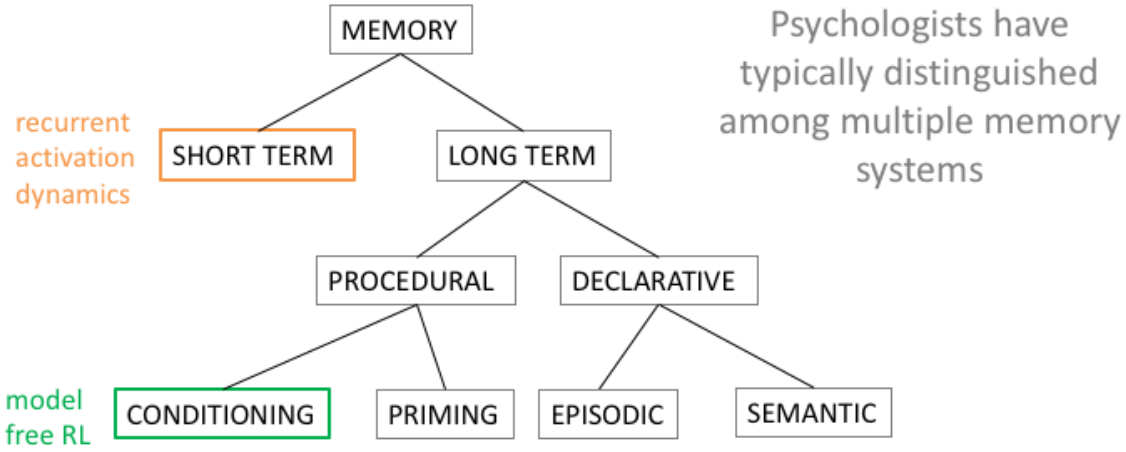
5.1. Gating in working memory systems



The parietal cortex is involved in integration of information over short timescales for rapid sensorimotor decisions

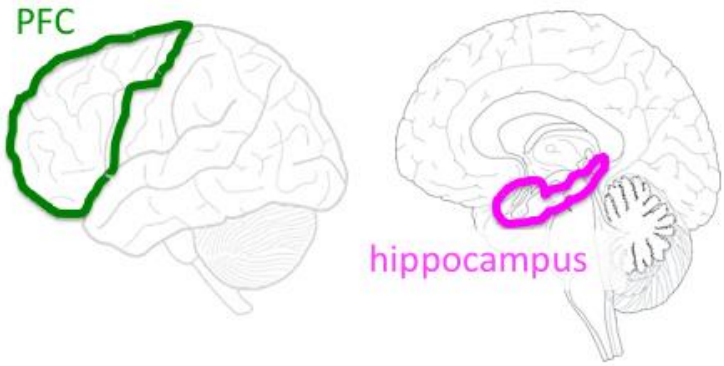
We have seen how RNNs may be a plausible model of integration in parietal cortex

Thus far we have seen that RNNs provide a powerful tool for action selection based on time-varying streams of data. We have also seen that they can be observed to predict the heterogeneity of coding properties observed in the parietal cortex during working memory and perceptual decision tasks. However, we have also discussed how RNNs involve computationally costly and biologically implausible training methods (BPTT), especially when a long history of stimulation must be taken into account to select a response. How is it then that humans are able to both learn and decide over multiple, often prolonged timescales?



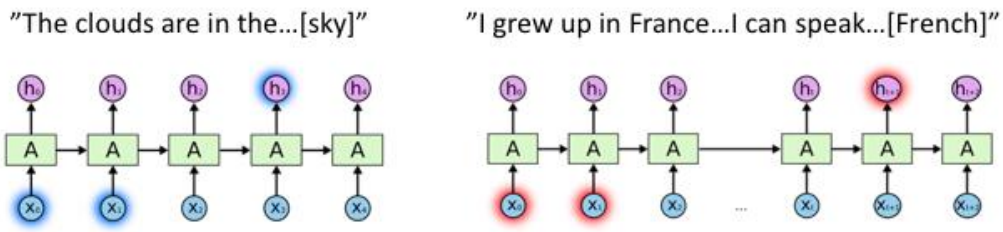
What computational problem have these memory systems evolved to solve?

A critical intuition is that humans (and many other animals) don't rely on a single memory system. Memory systems in most animals, and especially in mammals, are modular. Evolution has furnished animals with brain structures that allow information to be encoded, maintained and selected over multiple timescales. Above, I show a diagram that is often given on introductory psychology courses concerning human memory, which attempts to provide a taxonomy of human memory. Some of the modules seem to relate to processes we have already discussed. For example, "conditioning" is closely related to the concept of model-free RL (lecture 2) and the temporal dynamics of information integration was discussed previously (lecture 4) in the context of RNNs. Although we might reasonably dispute this taxonomy, the salient point is that there are many interesting and distinct memory modules. What computational problems have these multiple memory systems evolved to solve?



In particular, hippocampus and PFC are involved in more complex forms of memory and control

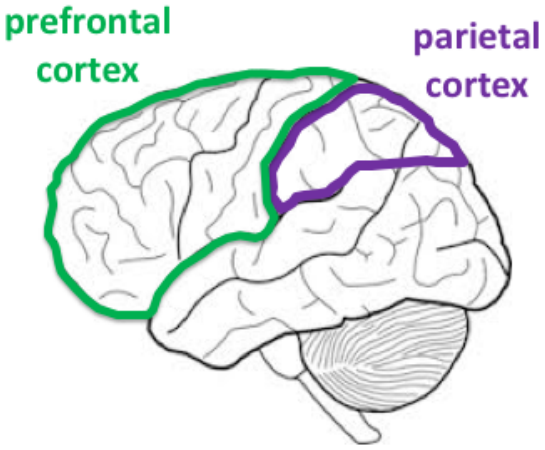
In the current lecture, we will focus on two structures – the prefrontal cortex and hippocampus – that are involved in more complex forms of memory and control. We will argue that these structures play an important role in learning and making decisions over multiple distinct timescales. We'll start with the PFC.



Recall the problem of BPTT: it's unfeasible except over relatively short timescales

Firstly, recall that in the previous lecture we discussed the limitations of backpropagation through time: although it works well over reasonably short timescales, such as those that might be relevant for rapid sensorimotor control, it's ineffective and/or implausible as a model for how information from the distant past can be brought to bear upon a decision.

The prefrontal cortex may be responsible for selecting outputs over multiple (longer) timescales, e.g. during language production and complex motor control

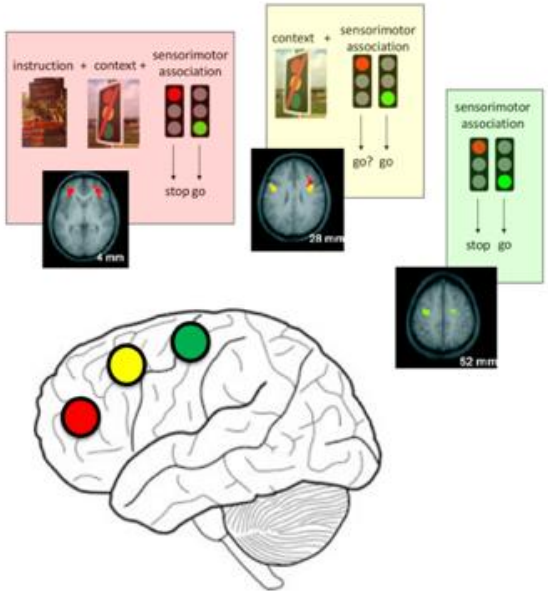


The parietal cortex is involved in integration of information over short timescales for rapid sensorimotor decisions

In particular, we are going to return to the proposal made earlier – that the brain involves multiple hierarchies of temporal integration, and that the integration window grows as one progresses along the rostro-caudal axis of the neocortex. We focussed on the hierarchy from sensory regions to the parietal cortex. However, we can extend this yet further. One might associate the parietal cortex (and some caudal prefrontal regions, such as the premotor cortex and FEF, with which parietal cortex is densely monosynaptically interconnected) with more rapid sensorimotor control.However, other more anterior portions of the PFC subserve action selection over much longer timescales, allowing more complex planning and reasoning processes that require extended maintenance and manipulation in working memory.

In support of this view, we know from classic neuropsychology that unilateral lesions to the parietal and premotor cortices lead to hemispatial neglect, where participants fail to engage in action selection towards the side of space contralateral to the lesion, indicative of a deficit of simple action selection. However, lesions to the anterior PFC lead to more subtle disruptions of behaviour, whereby motor control is unimpaired but patients display disordered planning and reasoning, as if they were unable to select behaviours according to a long-term goal.

5.2. Working memory gating in the PFC



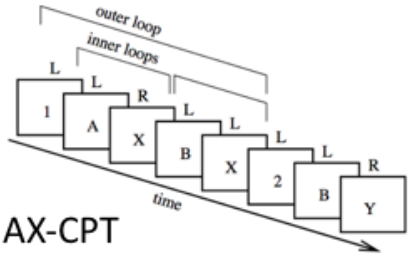
The **cascade model** of PFC control proposes that action selection occurs over multiple timescales

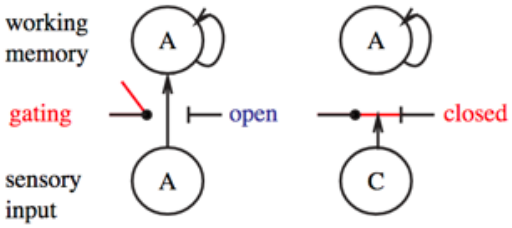
These are implemented in successively more rostral regions of PFC

Koechlin & Summerfield 2007

In fact, this theory that there is a gradient of action selection over multiple timescales has been formalised by various groups. One theory[51], known as the cascade model, is based on an information theoretic approach, suggesting that distinct loci within the frontal lobes – corresponding to premotor, caudal prefrontal, and anterior prefrontal sites – select actions that are conditioned on layers of context or instruction that stretch further and further back in to the past. We know that the premotor cortex is important for conditional action selection, such as when deciding to "stop" at a red traffic light and "go" at a green one. The theory suggests that caudal PFC activity (e.g. in BOLD) indexes the additional processing cost that is incurred when the action selection is further contingent on a contextual cue. For example, if there is a sign indicating that the traffic lights are out of order (e.g. because the road is being mended) the habitual response selection mechanism is overridden by an additional PFC signal that incorporates the context (traffic lights inoperational) into the decision. This is consistent with a long literature suggesting a key role for the PFC in suppressing a prepotent response in the service of controlled action selection. However, yet more anterior regions are required to incorporate decision-relevant information that may have been presented yet further in the past. For example, imagine that on an earlier encounter with a construction worker on the roadmending site had indicated that irrespective of what the "out of order" sign said, the traffic lights should be respected. This further information would need to be folded into the decision, even if it were received a considerable length of time ago. The most anterior regions of the PFC are required to incorporate this information into the decision. The cascade model of PFC function is supported by evidence from neuroimaging and lesion studies in which successive layers of instruction are provided in time that dictate the response required to a coloured cue, in a manner very similar to the traffic light example.

[51] Koechlin, E., Ody, C., and Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. Science *302*, 1181-1185, Koechlin, E., and Summerfield, C. (2007). An information theoretical approach to prefrontal executive function. Trends Cogn Sci *11*, 229-235.





If the last number was 1, press for AX
If the last number was 2, press for BY

Frank proposes that one outcome of striatal inputs to PFC is to **gate** an activity state in working memory circuits

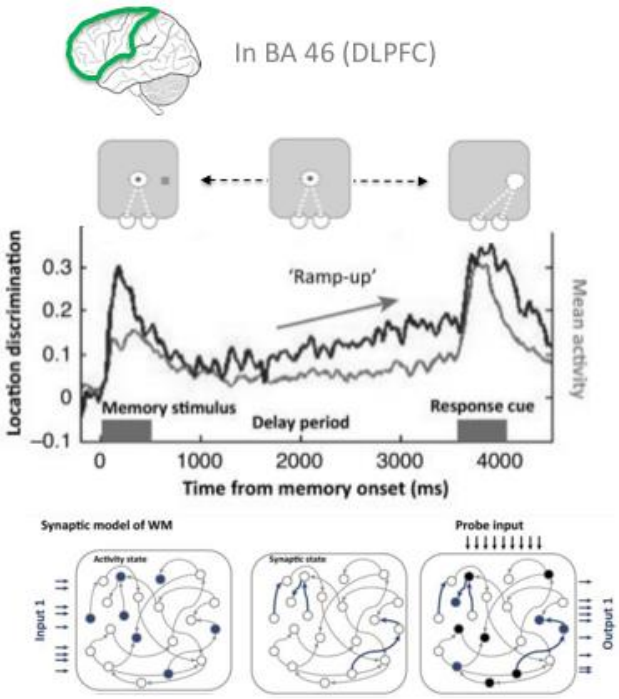
This allows the circuit to solve the **temporal credit assignment problem**, i.e. knowing which events caused an outcome

Frank et al 2001; Hazy et al 2006

It may be, thus, that whilst the parietal cortex and premotor zones engage in computations that are reasonably approximated by a vanilla RNN, a different mechanism is needed to account for integration of information over longer timescales, such as occurs in the primate PFC. What sort of computational mechanism might allow long-term, time-dependent action selection without incurring the computational cost of training an RNN? One class of solution involves the proposal that decision-relevant cues, rather than being persistently maintained by tonic activity (as in classic WM studies), causes a fast, temporary reconfiguration of synapses in higher cortical regions, that "gates" the information into working memory without requiring the maintenance of a prolonged activity state. Various proposals of this nature have been made, and the slide above illustrates one of the more successful. In the PBWM model proposed by Frank and colleagues[52], action selection requires the dynamic interaction between PFC and basal ganglia. When information is throughput from striatum to PFC, it engenders a fast plastic change at PFC synapses, which allows information to be gated temporarily into working memory. Note that here the claim is that working memory depends on synaptic change (e.g. a rapid update of the weights) rather than a persistent activity state. Subsequent cues (e.g. a contextual probe or "go" signal can open the gates, allowing the memory trace to re-enter the activation dynamics and go on to guide action selection. Frank and colleagues have shown how the PBWM model can explain how humans perform a contextual action selection task known as the AX-CPT, in which participants view a stream of numbers and letter and respond (or not) according to a complex rule depending on the stimulation history. In the AX-CPT, one of two conditional responses (e.g. respond to X after A but not B, or respond to Y after B but not A) is made according to whether the last number shown was a 1 or a 2, requiring participants to simultaneously maintain information in an "outer loop" (which was the last number) and an "inner loop" (was the last letter an A or a B). According to the PBWM model,

[52] Frank, M.J., Loughry, B., and O'Reilly, R.C. (2001). Interactions between frontal cortex and basal ganglia in working memory: a computational model. Cogn Affect Behav Neurosci *1*, 137-160, Hazy, T.E., Frank, M.J., and O'Reilly, R.C. (2006). Banishing the homunculus: making working memory work. Neuroscience *139*, 105-118.

each a number is gated into working memory and then retrieved only when required, i.e. to determine whether a response should be made to an X following an A.



In the synaptic memory model, the activity is maintained in fast plastic connections (or gates) rather than persistent activity states

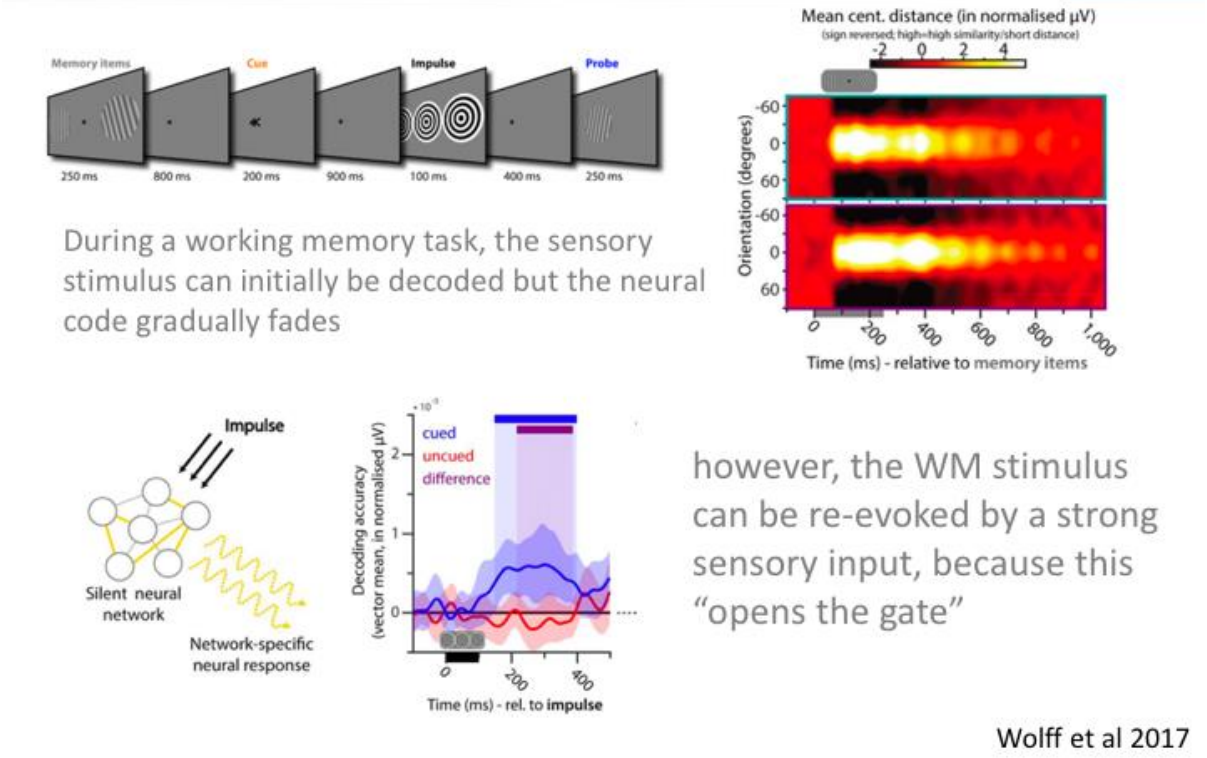
Predicts that random input to the system should re-elicit neural code for WM stimulus

Stokes et al 2015

The idea that working memory states may be mediated by fast synaptic plasticity is gaining traction in neuroscience. For example, this review article[53] offers an explanation for a range of past findings in nonhuman primate data that support this model. For example, the previously-described tonic firing in PFC during the delay period may be more related to future action selection than to maintenance, as it "ramps up" towards the eventual occurrence of the probe. Similarly, the presence of an irrelevant distracter stimulus occurring during the delay period induces only a momentary biasing of the single-cell responses, before they revert to coding the maintained stimulus, inconsistent with recurrent models of integration that rely on attractor dynamics, such the Wang model.

The authors also describe a new prediction: that random "pulses" of information passed through the system during the delay period should elicit the "activity-silent" states that are present in working memory, because information will flow through the reconfigured synapses and momentary transform the silent state in an active one.

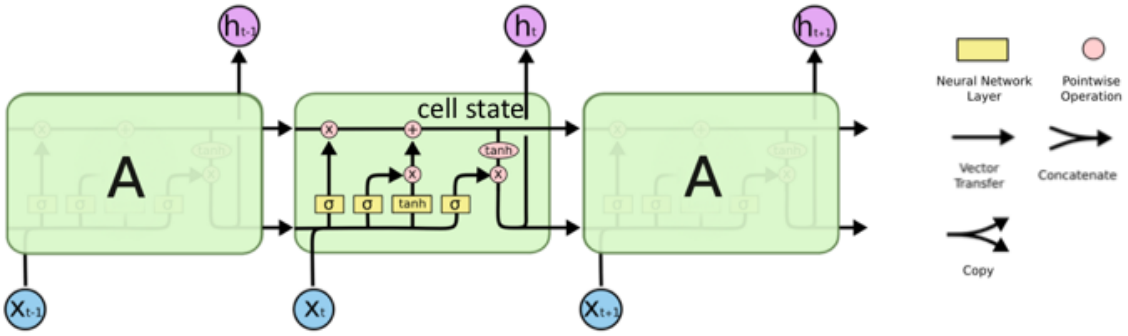
[53] Stokes, M.G. (2015). 'Activity-silent' working memory in prefrontal cortex: a dynamic coding framework. Trends Cogn Sci *19*, 394-405.



Consistent with this prediction, Wolff and colleagues[54] used multivariate methods to decode the contents (i.e. a grating) from EEG signals during the delay period of a retrocued working memory task. The decoded signal was strong following grating onset but died away across the retention interval. However, the sudden onset of a task-irrelevant ('pinging') stimulus that elicited a strong phase-locked signal (concentric rings) allowed the cued (but not uncued) grating to be momentarily decoded, as if the activity-silent state were momentarily reactivated. Similar results are obtained when the brain is "pinged" invasively using transcranial magnetic stimulation (TMS).

5.3. Long short-term memory networks (LSTMs)

[54] Wolff, M.J., Jochim, J., Akyurek, E.G., and Stokes, M.G. (2017). Dynamic hidden states underlying working-memory-guided behavior. Nat Neurosci *20*, 864-871.



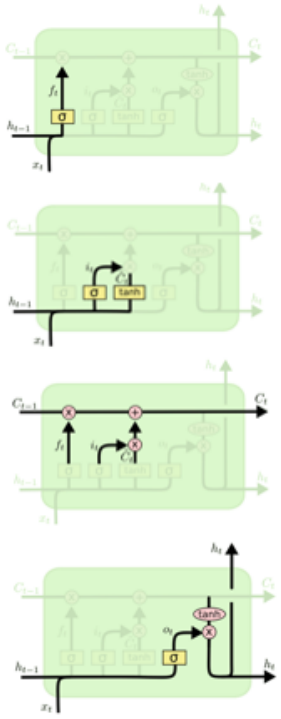
The full LSTM architecture involves distinct gates that switch states in or out of the hidden units

However, the principle is still the same, in that the hidden state is passed between cycles

Hochreiter & Schmidthuber 1997

Concurrently with this emerging view in neuroscience, a new neural network architecture has come to the fore that directly embodies the idea that information is "gated" into working memory in an activity-silent fashion. This architecture, first described by Jurgen Schmidthuber in the late 1990s, is known as a "long short-term memory network" or LSTM[55]. The architecture is a little complex, but it is a straightforward adaptation of the vanilla RNN network, but including new trainable weights that determine how information is gated into and out of a short term store, as well as which information in the store is overridden or forgotten.

[55] http://www.bioinf.jku.at/publications/older/2604.pdf



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) \qquad C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) \qquad o_t = \sigma(W_o \cdot [h_{t-1}, x_t])$$

$$\widetilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t]) \qquad h_t = o_t \times tanh(C_t)$$

Four distinct set of trainable weights control what is switched in or out on each timestep

Sigmoid nonlinearities determine how ”open” the switches are in range [0,1].

Hochreiter & Schmidthuber 1997

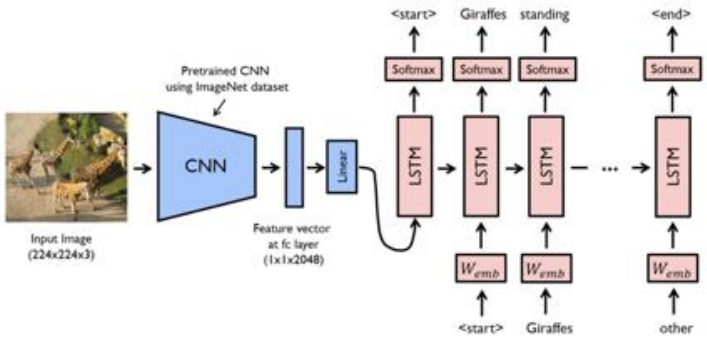
A comprehensive description of how an LSTM network functions is beyond the scope of this course, but there are numerous online resources providing a detailed explanation of its inner workings. Broadly, active information in the hidden layer flows through time as in a standard RNN, with the state on the previous timestep influencing that on the current timestep. However, there are now several interim states. Firstly, one set of trainable weights W_f determines which information is forgotten (or overwritten) in the hidden state. Other weights identify candidate sections of the activity-silent state to be reincorporated into the active state and (vice versa) determine what fraction of that information is reincorporated into the ongoing activity. Broadly, the LSTM learns to perform a gating function not entirely dissimilar to that proposed by theories of PFC function, such as the PBWM model.

One major use of RNNs and LSTMs is for image captioning

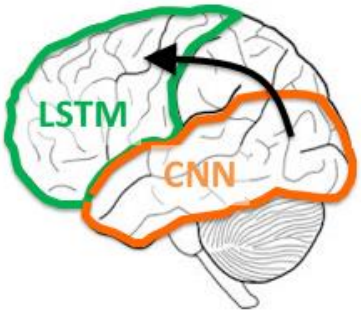
This allows the network output to be a sentence in natural language, rather than just a category label

Hochreiter & Schmidthuber 1997

LSTMs are even more powerful than RNNs and are now the tool of choice for researchers working with complex time-varying stimuli, such as natural language. One interesting use of these models has been in image captioning. Rather than simply being trained to predict a class label associated with an image, the network is trained (with supervision) to produce a sentence describing what is present in the picture. Some examples are shown above.



High-performing networks often involve a CNN that sends its output to an LSTM

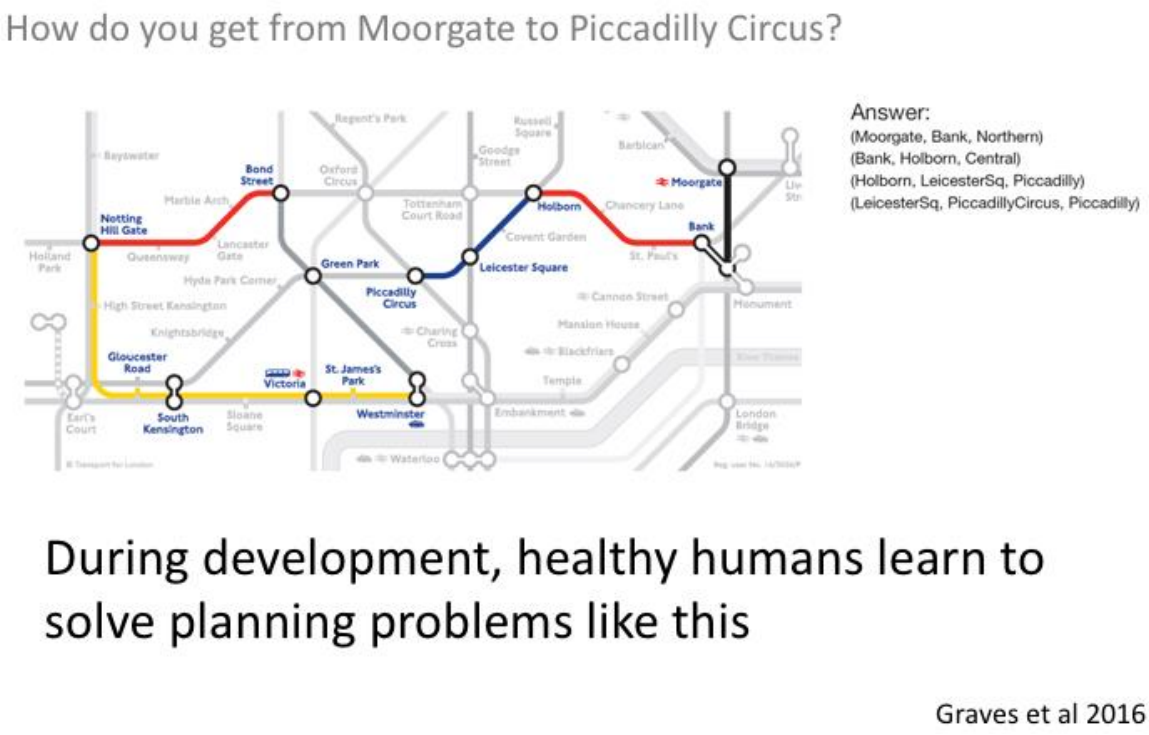


Just as sensory signals are sent to higher association cortex for complex decisions

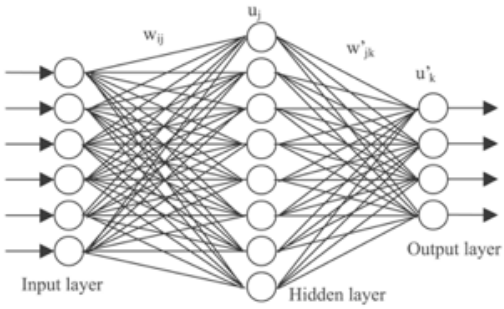
But of course, the PFC does much more than WM maintenance and gating…

It's interesting to compare the overall architecture of these image captioning models with our global understanding of how brain function is organised in the neocortex. For example, many neural network models incorporate a CNN on the front end (to disentangle the image pixels into sensible representations), whose output is then passed into various LSTM layers to be combined with a sentence in natural language. Compare with our understanding of how the brain deals with images (in the ventral stream), with information then throughput to the prefrontal cortex (dealing with, for example, speech comprehension/production). This is what I meant when I claimed that it is researchers in AI/ML that are formulating (and implementing) *general theories of brain function*, i.e. starting to think about how to wire up all the various computational components required to build a brain that produces intelligent behaviour.

5.4. The Differentiable Neural Computer



However, of course, we know that the PFC (and parietal cortex) do much more than working memory maintenance. For example, the PFC seems to be instrumental for planning and reasoning in complex domains. Without your PFC, you are likely to be impaired at planning a new route through a complex environment such as the London Tube system – and much more at planning a route through a less familiar environment (such as the Shanghai Metro system).

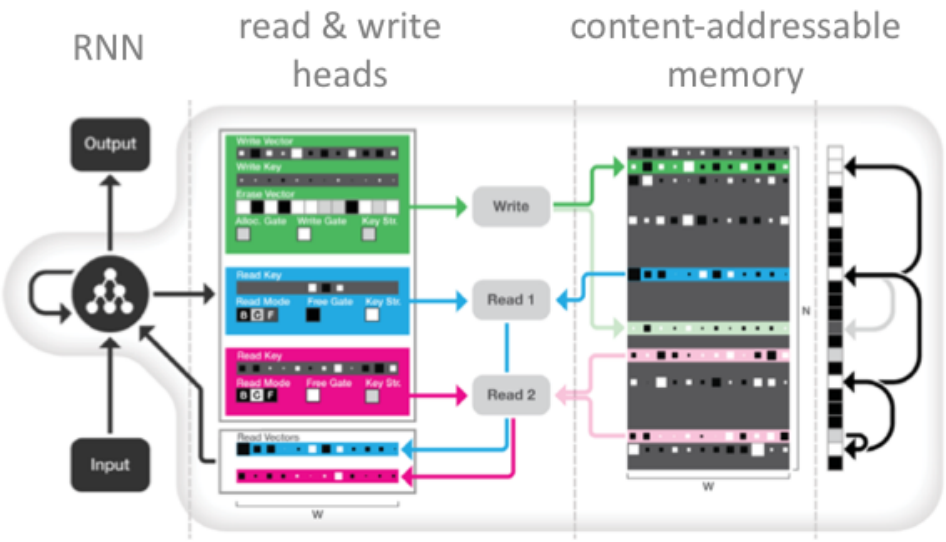


Computers separate computation (in the processor) and memory (on the disk).

In neural networks, computation is memory and vice versa

It's interesting that this class of problem involves precisely the sort of means-end reasoning the symbolic AI approaches were designed to solve. However, recall that the limitation of these systems is that researchers had to "build in" the reasoning policies required to solve the problem – in other words, the system did not learn to reason, they reasoned according to a system that was handcrafted by the researcher. This limited the flexibility of the systems to cases where the symbols were uniquely specified by the experimenter (and not learned from the environment).

However, symbolic systems had an interesting feature, in that the computational operations were separated from the contents over which they operated. Thus, a reasoning system operates according to a set of (pre-specified) logical rules; the researcher can choose to input any set of relevant inputs. Thus, in a sense, these systems separated computation (the operations of the processor) from memory (the inputs on which computation operated). This is also a hallmark of most modern desktop computers, where the processor (CPU) and the memory (e.g hard disk) are distinct computational components. In neural networks, by contrast, memory and computation are the same thing. The computations are determined by the values of the network weights, which *are* the memories that the network retains.

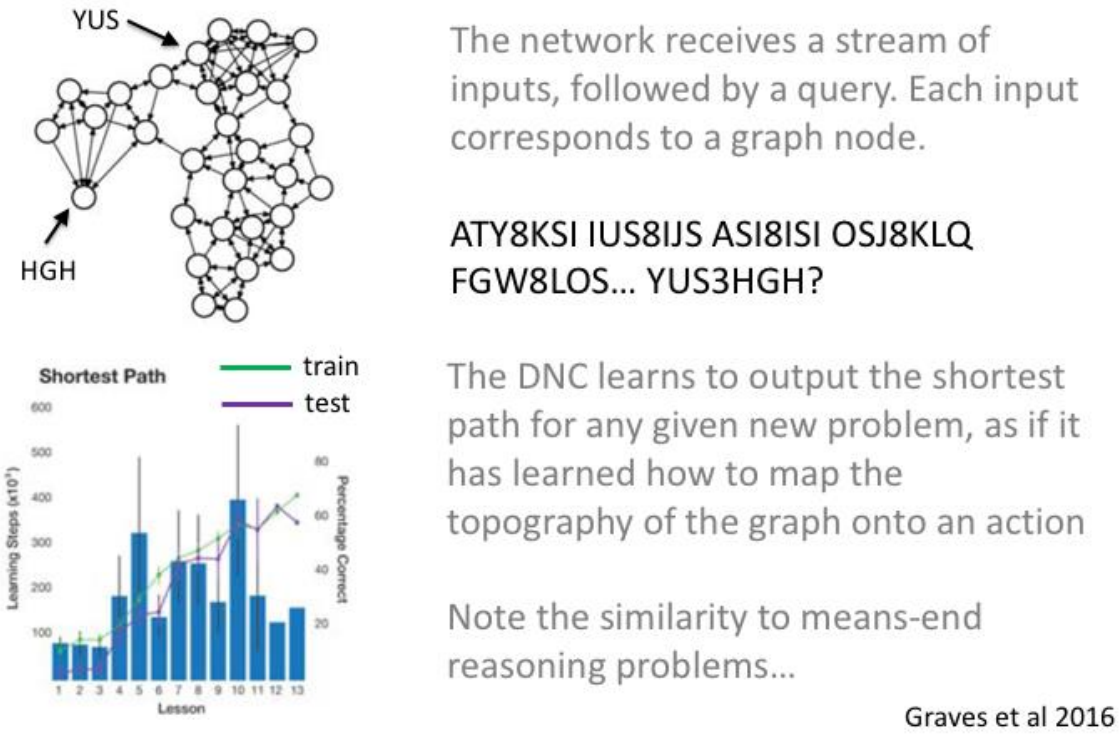


The DNC uses an RNN controller that learns to write information to a content-addressable store

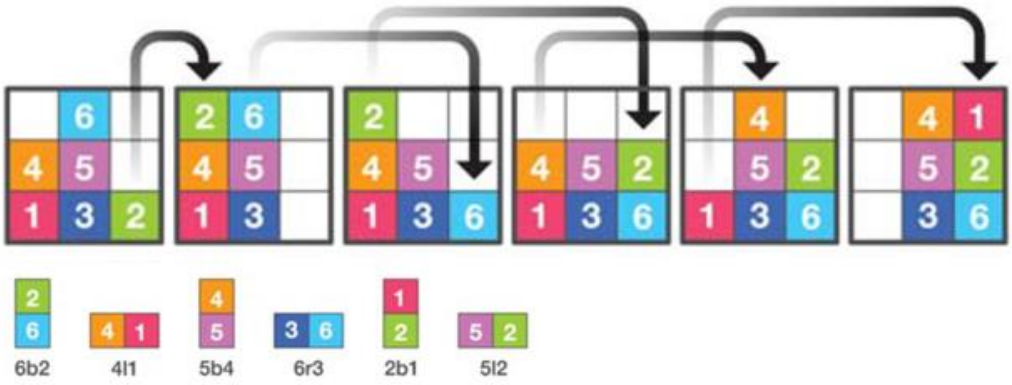
Graves et al 2016

So can one build a neural network which embodies the best of both worlds – in which reasoning policies are learned by experience (i.e. via "end-to-end" training, for example with gradient descent) but there is a separate memory store which codes the contents over which planning or reasoning policies operate? In 2016, Graves and colleagues[56] described one such architecture, which they called the "differentiable neural computer" or DNC. Its basic architecture is shown on the slide above. At the heart of the network is an RNN that acts as a controller. However, rather than acting to open or close gates in working memory, it acts to write information to and read it from a content-addressable store, like a long-term memory system.

[56] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., *et al.* (2016). Hybrid computing using a neural network with dynamic external memory. Nature *538*, 471-476.



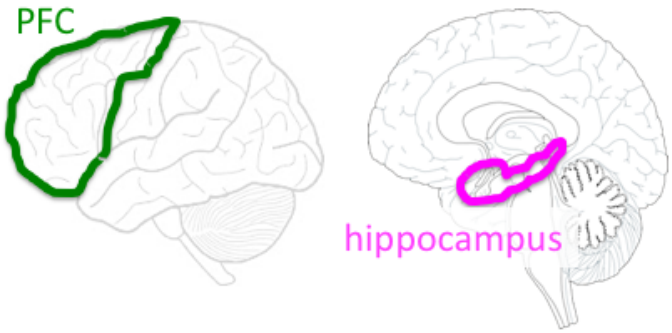
Here's an example of the sort of task the DNC was asked to solve. It's the type of supervised learning problem that might typically be tackled with an RNN or LSTM, but as we shall see, these simpler architectures perform more poorly on the tasks employed. The network receives a long stream of character triplets that denote the structure of a graph, followed by a query and two triplets indicating a start and goal location in the graph. For example, it's as if you were asked to navigate from "HGH station" to "YUS station" in an alien subway network. The network is trained (with supervision) to output the shortest path from HGH to YUS as a stream of intervening nodes. This doesn't sound like a very hard problem, until you realise that the researchers input a completely different graph on every training step, so that the network wasn't just learning a fixed policy (like a set of principles for navigating the Tube) but a *general policy* for finding the shortest path for a graph with any topographic organisation. Training the network to solve this is hard, but the researchers succeeded, in part by using a "curriculum" that started with smaller, simpler graphs that gradually increased in complexity.



The network can also solve SHRDLU-type problems, but now learning from end-to-end rather than using hard-coded logical reasoning

Graves et al, 2016

In a nice parallel to the earlier GOFAI work, the authors also trained the network to solve a version of SHRDLU, the blocks world problem first used by Terry Winograd. The network was able to solve complex reasoning problems with stacks of blocks, but rather than using a handcrafted policy, it learned to reason about the blocks entirely via supervised learning, solving complex problems related to the Tower of Hanoi task requiring up to 10 planning steps.



Neurally, to what might the DNC correspond, if anything?

Candidates clearly include the PFC, where there exist long-term representations of task-relevant variables. But the rapid writing to content-addressable memory also sounds like the hippocampus…

In the next lecture, we will discuss how hippocampal and neocortical systems might interact to solve multiple tasks that occur in a temporal sequence

It remains unclear whether the DNC is a plausible model for neurobiology at all, and if it is, whether it more closely resembles the PFC or the hippocampus. Other large-scale models that incorporate a content-addressable memory are being developed – these are sometimes referred to as "world models".

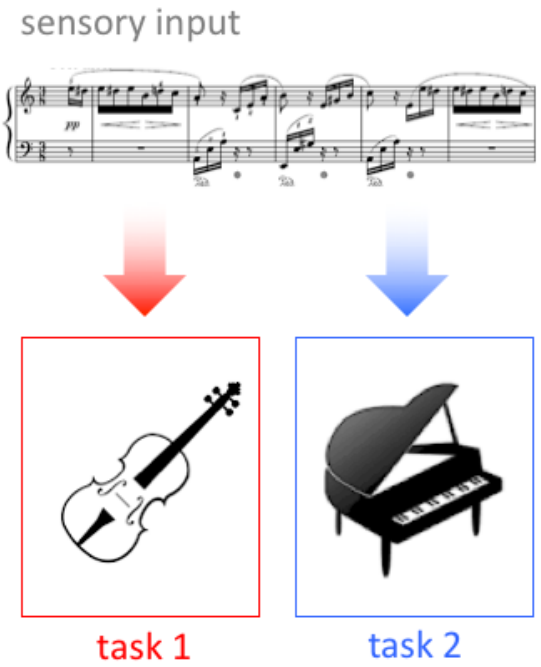
5.5. The problem of continual learning

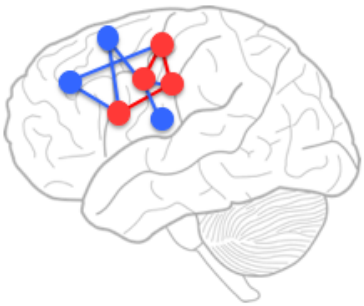
Humans and other animals continue to acquire new knowledge over the lifespan

LSTMs and the DNC are candidate solutions to a critical problem: how to bring information that may have been acquired in the potentially distant past to bear upon a decision. These classes of network solve this problem by using putative storage mechanisms – underpinned by either fast gating mechanisms, or a content-addressable memory – to maintain information about past states in a way that is protected from interference by current computation. Both classes of model draw in some way on our understanding of the storage mechanisms in human memory, and in particular on new research into the neurobiology of working memory in the PFC and/or hippocampus.

However, a distinct but related problem is how to *learn* over multiple timescales. Animals begin to learn when they are born and are able to continue to acquire new information well into adulthood and old age. This is extremely important, as it allows agents with long lifespans to acquire a rich knowledge of their environment – the "wisdom of the elders".

Thus far in this course, we have considered a number of applications for which machine learning systems offer good (or even superhuman) performance, such as image classification. But here instead we are going to focus on an important limitation of current ML systems. Currently, the problem of how to learn over the lifespan is unsolved in AI research, but it is becoming increasingly clear that the organisation of modular memory systems, and in particular the hippocampus, may play a critical role in permitting this "lifelong" or "continual" learning.

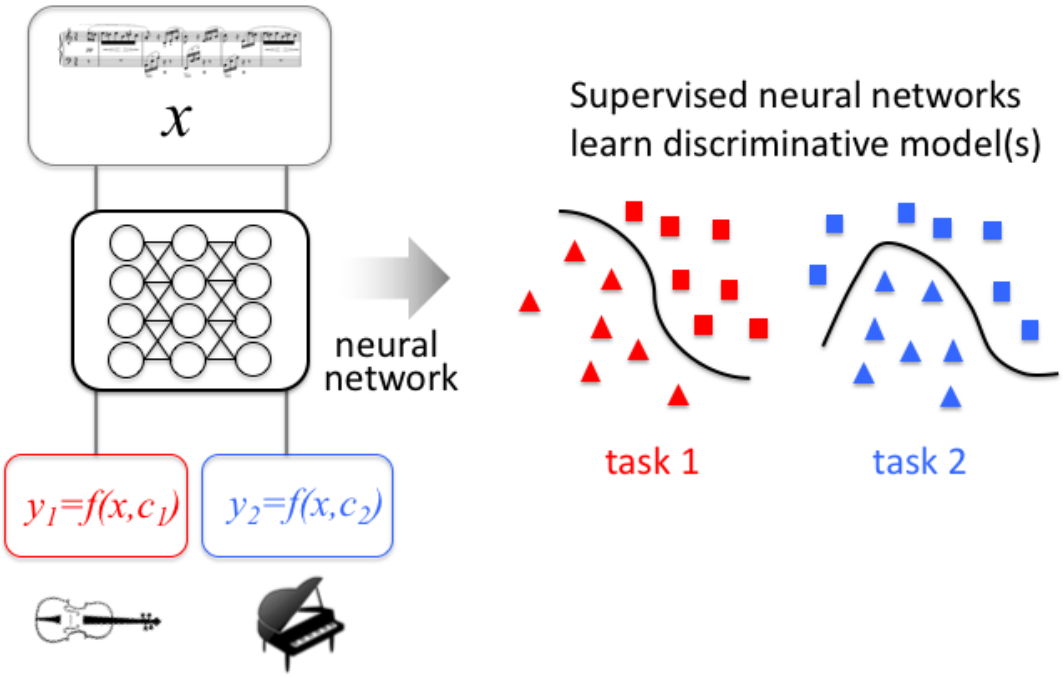




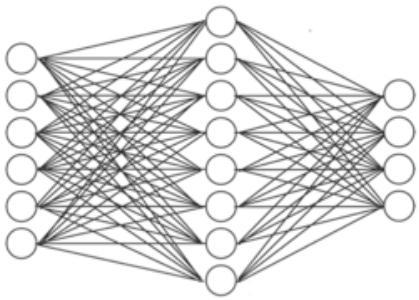
Consider the problem of learning two distinct tasks, such as playing the piano and the violin

To illustrate the problem of "continual" learning in a simple setting[57], imagine you are a musician who is learning to play both the cello and the violin. You might practice the cello in the morning, and the piano in the afternoon. Importantly, your afternoon practice session does not erase the memory of everything that you've learned in the morning! When you wake up the next morning, your performance should be incrementally better on both tasks – so that after a lifetime of practice, you might become virtuoso on both instruments. As we shall see, this is not the case for current ML systems, in particular feedforward networks that are trained slowly with gradient descent alone.

[57] The most useful paper for all of what follows: Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. McClelland JL, McNaughton BL, O'Reilly RC. Psychol Rev. 1995 Jul;102(3):419-57. See also this more recent update: Complementary learning systems. O'Reilly RC, Bhattacharyya R, Howard MD, Ketz N. Cogn Sci. 2014 Aug;38(6):1229-48.



So how can we think about the cello/piano problem from perspective of a feedforward network? Well what we want is one network that can learn two functions: $y_1 = f(x, c_1)$ and $y_2 = f(x, c_2)$ where c_1 and c_2 are different tasks that may be performed in distinct contexts (i.e. learn cello, learn piano). In the case of supervised learning, we can think of the two functions as offering two different discriminative functions for classifying the same data x (although a comparable problem arises when learning two discriminant functions for data drawn from different distributions).



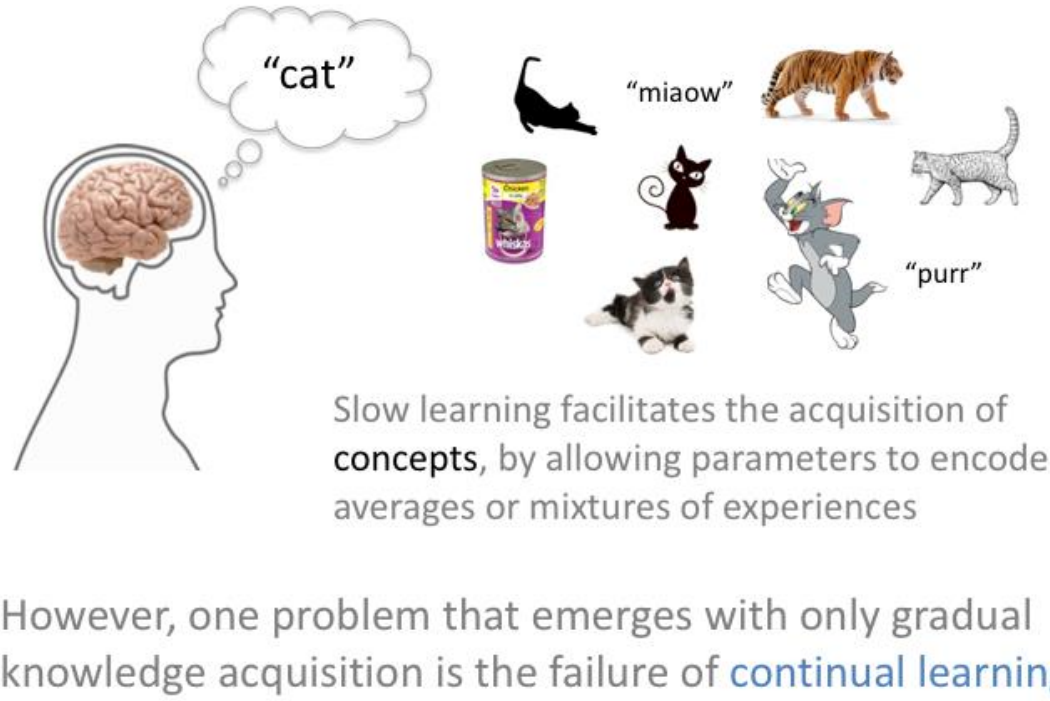
Weights are updated by derivatives × learning rate α, where α is typically small e.g. < 0.001

In neural networks, learning occurs only gradually, via incremental changes in connections

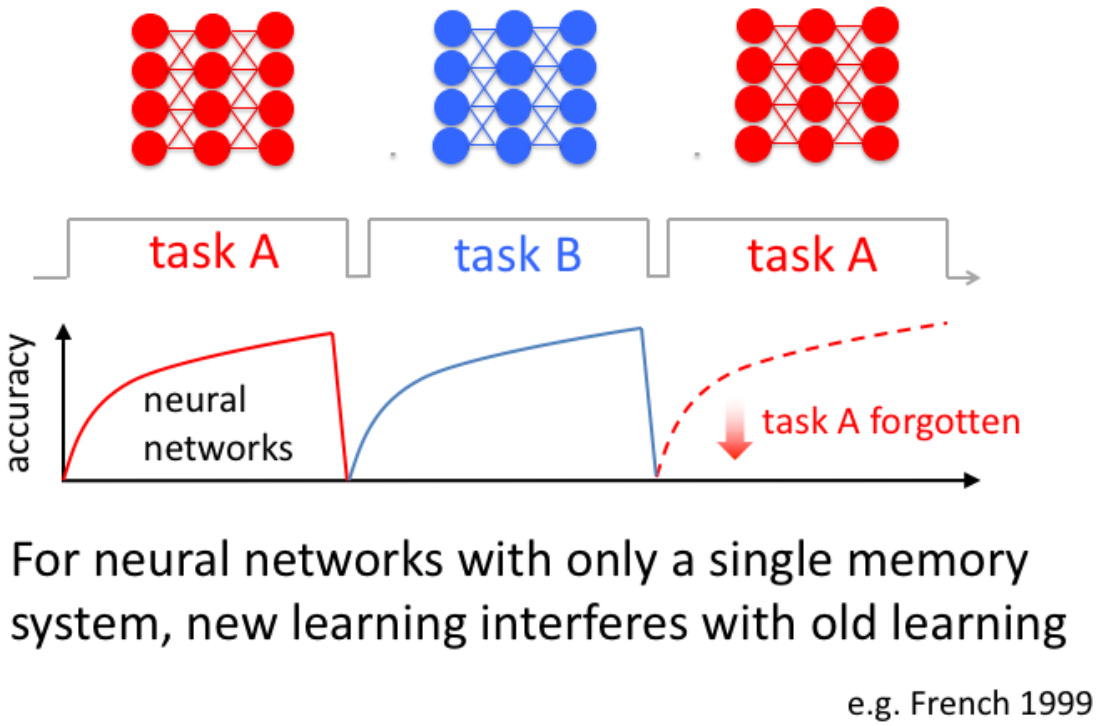
This has costs and benefits

We have learned that neural networks are powerful function approximators. As long as the network has sufficient training capacity, why should this situation present a problem?

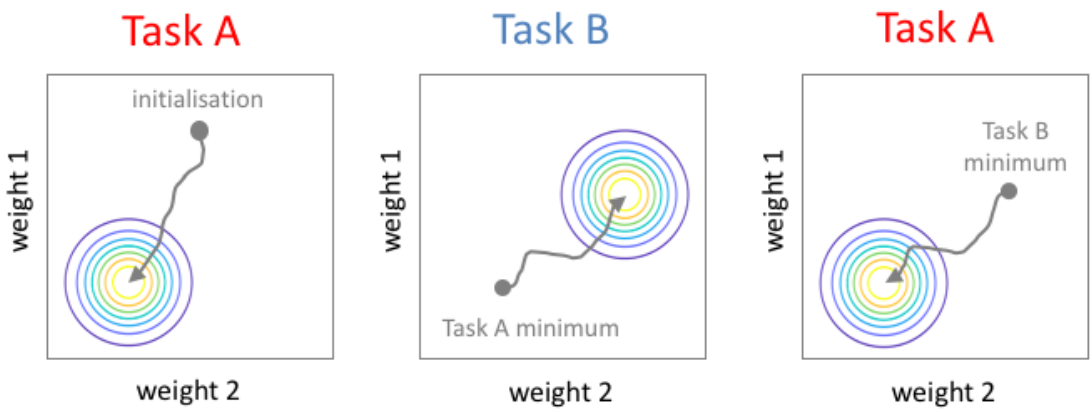
Recall that the rate at which optimisation proceeds is governed by the learning rate α, which dictates the step size by which the weights are updated according to the gradient of the loss. As we have seen, values of alpha that are too large and likely to lead to failures of convergence, and so small learning rates are often required. The incremental learning which is the hallmark of deep networks has both costs and benefits.



The major benefit is that slow learning allows the network to form representations that depend on a larger sample of the training distribution, rather than (for example) just a single item. To understand why this is the case, consider what happens if the learning rate is 1 (apart from the fact that your network fails to converge). With α = 1, the network fully updates the weights towards the target after every single sample, and it never takes more than a single item into account when learning – on the next trial, it simply forgets about all previous items and learns according to the very latest feedback it receives. So in this setting, the network cannot learn to aggregate over the information that may be present in multiple sequential samples. By contrast, low learning rates allow the network to form representations that depend on multiple training examples and the network can thus "abstract" over those training examples to form rich conceptual representations. Thus, the network can learn that the concept of "cat" is associated with cats of different breeds and sizes seen from different orientations, or with inputs across multiple modalities (miaow). By forming representations that depend on the overall statistics of the training examples, rather than a specific instance, the network is able to generalise to novel examples with comparable statistics.



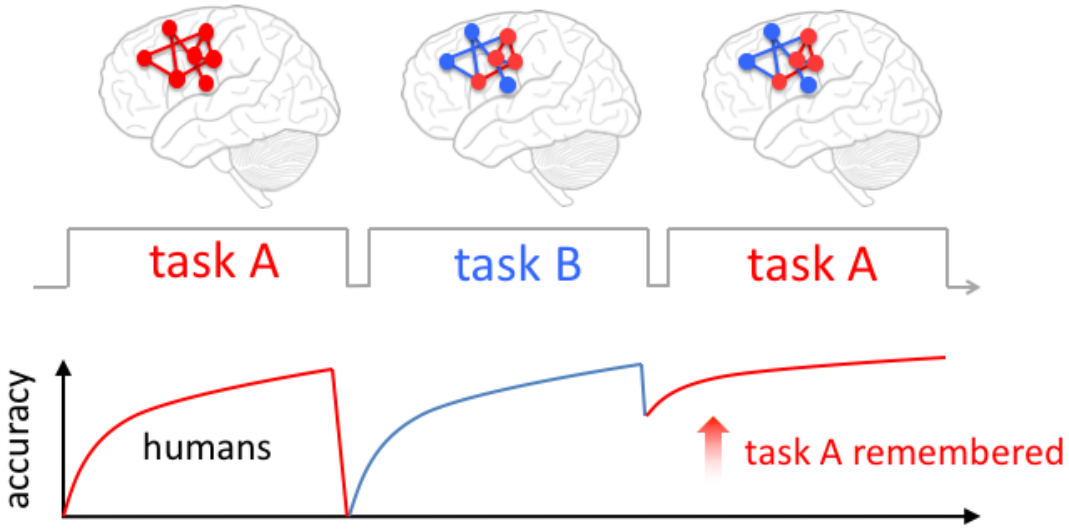
However, the cost of using exclusively a slow learning system is that it is very hard to learn continually, i.e. to perform one task for a prolonged period and then switch to perform another. Rather, standard neural networks suffer from "catastrophic interference". Consider a network that performs task A (e.g. play the cello) for a prolonged period and learns it to convergence. Subsequently, the network encounters a new task B (e.g. play the piano). The network can also learn this task to convergence. However, when returning to task A, the new learning (B) has "overwritten" knowledge of how to perform task A, and so the network has to relearn from scratch. Clearly, this is very different from the behaviour of (say) a human musician.



During the learning of task B, the weights shift to a new minimum, and when A is reintroduced, they have to shift back

e.g. French 1999

To understand why this happens, consider what changes occur in the weights. For illustration, the settings of just 2 weights are plotted on the slide; but the same principle holds for the higher-dimensional setting of a standard neural network. The weights are initialised at random and during optimisation for task A, they gradually shift to one of potentially many global minima that allow task A to be performed effectively. Following the introduction of a new objective B, the weights shift away from this point and towards a minimum that allows effective performance of task B. Thus, following the reintroduction of task A, they need to shift back again. All this happens slowly an inefficiently, because of the single (low) learning rate that determines the speed of learning.

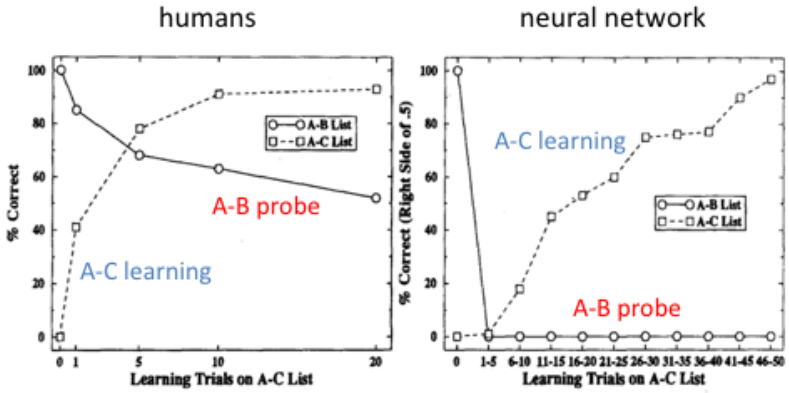


Humans typically exhibit much less interference

Of course, for humans a very different pattern is typically observed, whereby learning task B might cause a small amount of interference. However, in general, for human relearning on task A will start from a higher baseline and/or proceed much faster than initial learning.

A-B list
cat-house
bicycle-field
fish-car

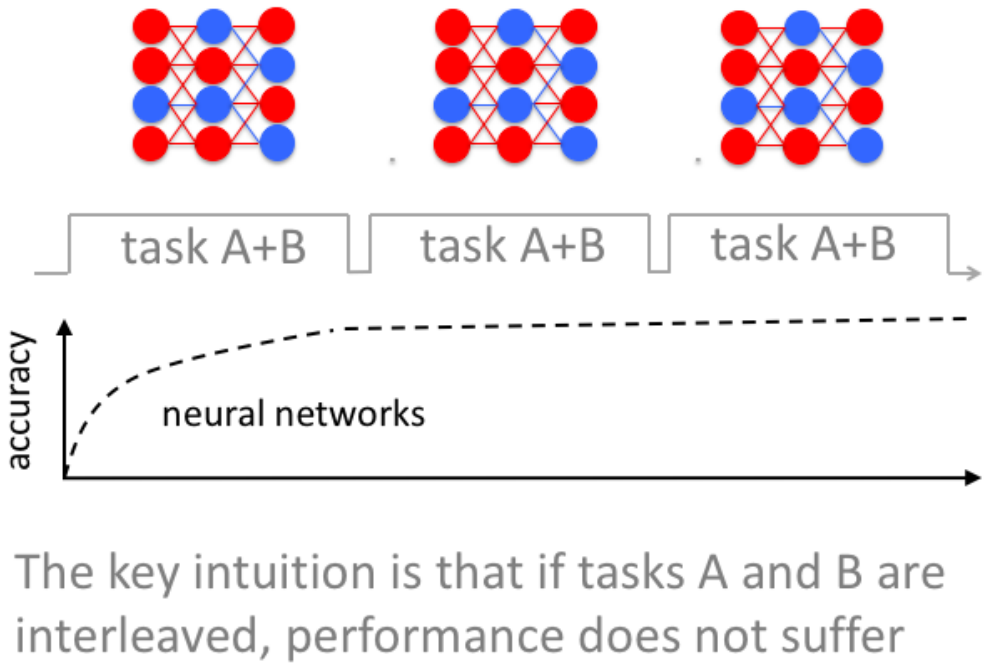
A-C list
cat-clock
bicycle-road
fish-cloud



In paired associate learning, humans show less interference for old associations during new learning (c.f. massed vs. spaced training)

McCloskey & Cohen 1989

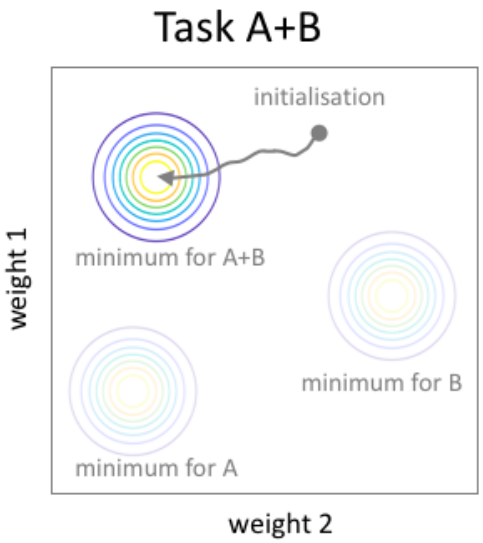
This contention has empirical support. For example, in this classic study by McCloskey & Cohen[58], human participants first learned a list of paired associates (A-B list), and then learned a new set of associates for A (A-C list), whilst being constantly probed for recollection of the A-B list without any feedback. Performance on the A-B list was impaired by A-C learning, but not nearly as dramatically as in neural network simulations, where A-B learning rapidly fell to the floor as the network started to learn A-C.



You might be thinking: well how do we know that the network actually has enough capacity to learn both tasks? Perhaps it only has sufficient memory to learn task A or task B, but not both?

For most settings, this is not the case. Indeed, if the two tasks are trained *together*, so that they resemble a single "macro-task", then the network is perfectly able to learn both A and B together. In other words, the problem of continual learning occurs in a "blocked" setting, where each task occurs over a prolonged period before switching to the next. This tends to be the case in natural environments. However, for neural networks, the problem is mitigated in "interleaved" environments in which all training samples are randomly intermixed. This is because without further architectural constraints, the network does not have a mechanism for partitioning knowledge in a way that prevents mutual interference among tasks. In fact, it doesn't really know what a "task" is at all. Rather, it just learns a conditional mapping from inputs to outputs; it treats all inputs alike as if they were part of a single, global task.

[58] McCloskey M & Cohen NJ (1989). Catastophic Interference in Connectionist Networks: The Sequential Learning Problem. Journal of Learning & Motivation, 24:109-165.



With enough capacity, the network **can** jointly solve tasks A and B, but it needs to find the correct minimum

Neural networks are **overparameterised** (many possible solutions could be found) due to large number of parameters

The solution found depends on the **initialisation**

Problem of intelligence: find the global minimum for all possible tasks!

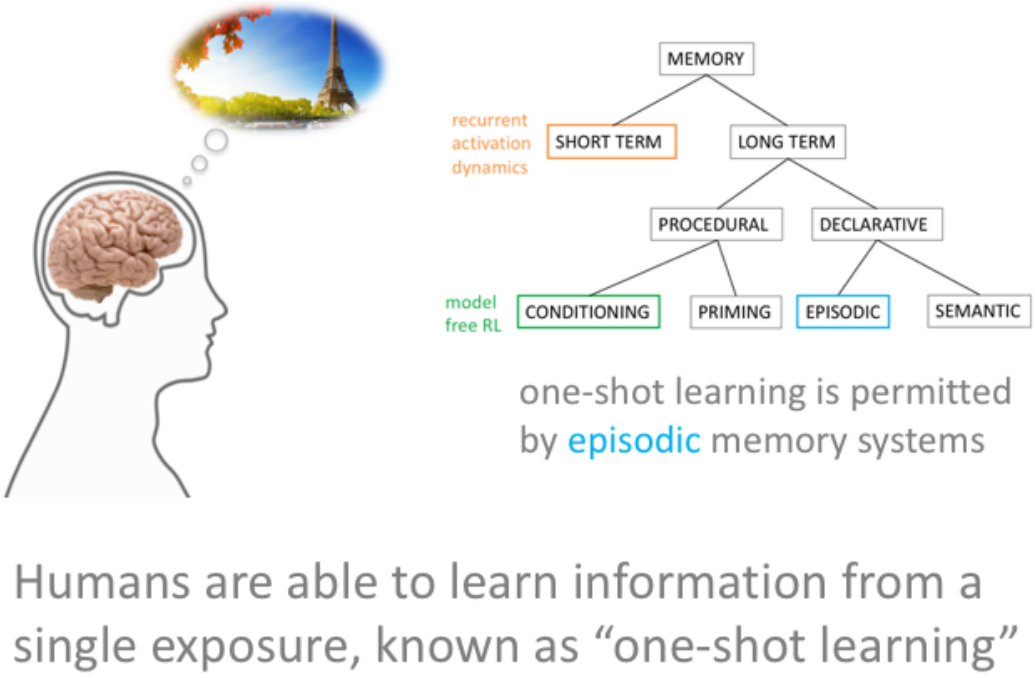
One reason why "interleaved" training tends to be successful is that neural networks tend to be overparameterised. That ensures that there exist multiple possible weight settings (minima) that allow task A and B to be solved, including settings that allows both A and B to be solved at the same time. So when training on A and B together, the network gradually converges to one of these "joint" solutions. Unfortunately, there is no guarantee that when training on A the network will also find a solution that allows B to be solved, and so during a "blocked" training setting the network oscillates back and forth between solutions that allow only one task to be solved at a time. This is why catastrophic interference tends to occur.

Thus one could think of the problem of building an agent that is able to perform any task, just like a human, as identifying a training regime that ensures that the network converges to just the right global minimum, where all relevant tasks can be solved! Much contemporary machine learning research implicitly buys into this view. However, this is clearly something that can be discussed. For example, an alternative is that humans may have "metalearning" mechanisms that allow them to "learn how to learn", so that they can rapidly adapt to any new task, without there being a single fixed global minimum that is sufficient for effective performance across an environment constituted by a distribution of tasks[59].

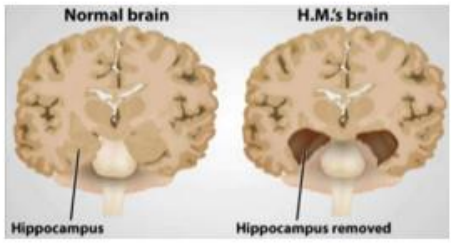
[59] This is an extremely promising approach which can be read about here: Prefrontal cortex as a meta-reinforcement learning system. Wang JX, Kurth-Nelson Z, Kumaran D, Tirumala D, Soyer H, Leibo JZ, Hassabis D, Botvinick M. Nat Neurosci. 2018 Jun;21(6):860-868.

6. Complementary Learning Systems Theory

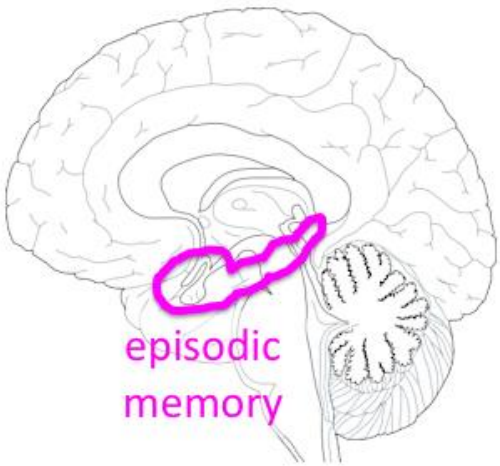
6.1. Dual-process memory models



So how is it that humans are able to avoid catastrophic interference when neural networks are not? One very critical aspect of human memory is the ability to vividly recall past experiences after only a single exposure, such as when you remember your experiences on a past holiday. In our taxonomy of human memory systems, this is known as episodic memory.



Hippocampal lesions provoke anterograde amnesia, i.e. failure of episodic encoding



One-shot learning (or episodic encoding) depends on medial temporal lobe structures such as the hippocampus

Classic theories in neuroscience and psychology ascribe this ability to the functioning of the hippocampus. Patients with damage to the hippocampus fail to acquire new information after a single exposure (known to AI researchers as "one-shot" learning) but are still able to acquire new skills gradually. For example, HM could learn new motor skills[60], and exhibit perceptual and semantic priming, but was unable to recall events or experiences that occurred after surgical removal of his medial temporal lobes (including the hippocampus).

[60] First described by Milner: Milner, B. (1962). Les troubles de la memoire accompagnant des lesions hippocampiques bilaterales. In Psychologie de l'hippocampe. Paris: Centre National de la Recherche Scientifique. If you want a version in English, try this more recent paper: Intact acquisition and long-term retention of mirror-tracing skill in Alzheimer's disease and in global amnesia. Gabrieli JD, Corkin S, Mickel SF, Growdon JH. Behav Neurosci. 1993 Dec;107(6):899-910.

This theory builds on a long tradition distinguishing recollection and familiarity in memory systems

For example, hippocampal amnesics learn fear associations without explicit memory for the fear-inducing event; exhibit intact priming

Claparède's drawing pin

In healthy humans, there is a (controversial) distinction between "recollection" and "familiarity" in the memory literature (c.f. the "butcher on the bus" phenomenon)

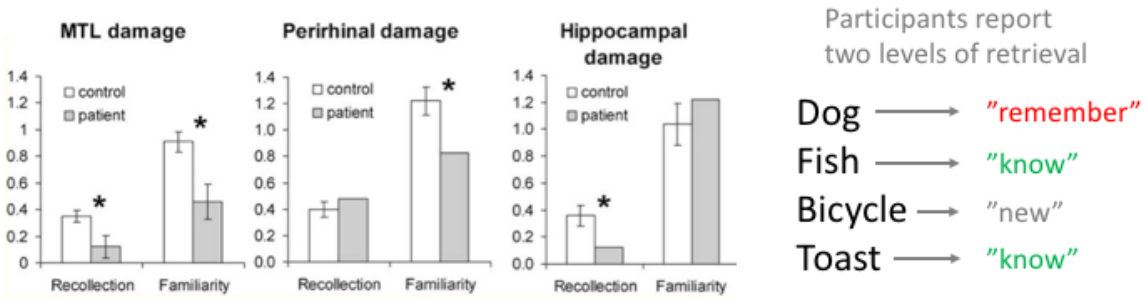
Yonelinas, 2002

This functional distinction between the memory systems in hippocampus and cortex has an established tradition in psychology and neuroscience, and indeed relates back to the notion of "implicit" and "explicit" memory that was first proposed at the beginning of the cognitive revolution in the 1970s. More recently, "dual-process" memory models have argued for a distinction between two classes of retrieval event: "recollection" (putatively hippocampal-dependent) and "familiarity" (putatively cortical)[61]. The key proposal is that recollection and familiarity are dissociable retrieval processes (the counterproposal, most often associated with Larry Squire, is that familiarity is a weak form of recollection). The evidence for the former theory is that it seems to be possible to have a strong sense of familiarity with a memory item without recollecting the relevant contextual information – i.e. where it was experienced and other associated detail. Anecdotal evidence for this comes from the "butcher on the bus" phenomenon, whereby people sometimes report meeting an acquaintance whom they recognise strongly without being able to place them, i.e. to remember their name or the context in which they are known. Lab-based experimental evidence, based on subjective reports in the "remember/know" paradigm, supports the idea that memory items can be highly familiar but retrieved without associated contextual information (i.e. not recollected).

Further supporting the dissociation between familiarity and recollection, amnesic patients such as HM often show evidence of familiarity without conscious recollection. For example, in the famous anecdote of "Claparède's drawing pin", the malicious Swiss neurologist[62] shook the hands of amnesic patients with a drawing pin concealed in their palm, leading the patient to withdraw the hand after receiving a nasty prick. The patients subsequently exhibited no recollection of this event but would refuse to shake Claparède's hand any more.

[61] Try this: Yonelinas, AP (2002). The Nature of Recollection and Familiarity: A Review of 30 Years of Research. Journal of Memory and Language 46(3):441-517

[62] https://en.wikipedia.org/wiki/%C3%89douard_Clapar%C3%A8de

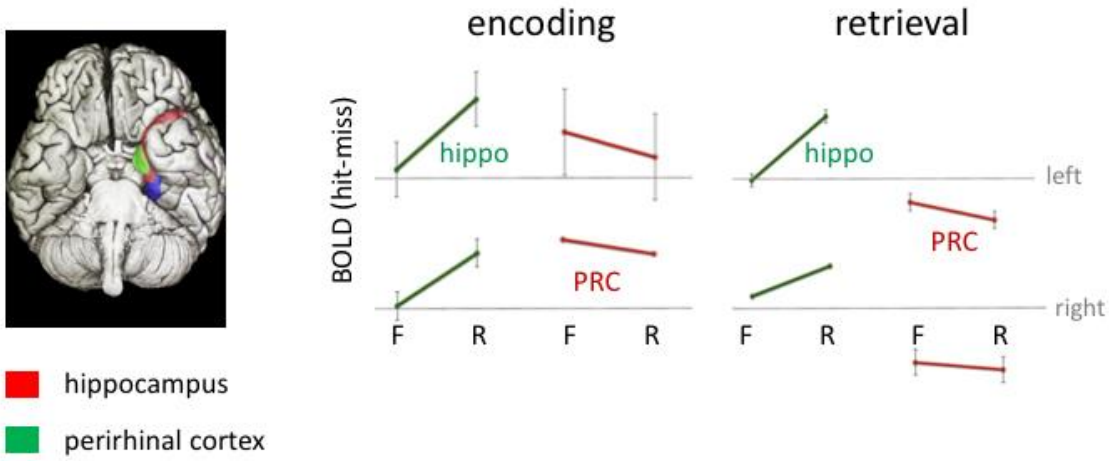


Lesion studies suggest that recollection depends on the hippocampus, familiarity on the cortex

Yonelinas, 2002, 2010

Indeed, dual-process memory models are supported by dissociations in the behaviour exhibited by patients with damage to hippocampus and nearby neocortical regions, such as the perirhinal cortex. After having learned a list of words, patients are shown both old words and new words (lures) and asked to subjectively report whether they "remember" or "know" the word. Hippocampal patients tend to exhibit deficits of recollection (remembering) with spared familiarity (knowing) whereas the reverse is true for patients with perirhinal damage. By contrast, patients with medial temporal lobe (MTL) damage encompassing both regions are impaired on both recollection and familiarity[63].

[63] Yonelinas AP, Aly M, Wang WC, Koen JD. Recollection and familiarity: examining controversial assumptions and new directions. Hippocampus. 2010 Nov;20(11):1178-94.



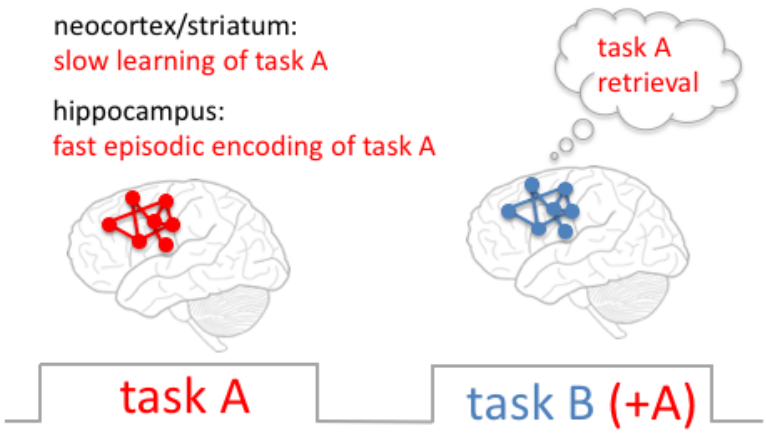
Converging evidence from meta-analysis of ~17 fMRI studies

Wais, 2008

There is also converging evidence from fMRI studies for this "remember" vs. "know" distinction in the hippocampus/neocortex. For example, in this metanalytic study[64], BOLD signals were consistently higher in the hippocampus when participants reported "remembering" an item (i.e. recollection) but higher in the perirhinal cortex when they reported "knowing" an item (i.e. familiarity).

6.1. The hippocampus as a parametric storage device

[64] Wais PE. FMRI signals associated with memory strength in the medial temporal lobes: a meta-analysis. Neuropsychologia. 2008 Dec;46(14):3185-96



CLS proposes that hippocampal-dependent memories are constantly interleaved with ongoing learning

McClelland et al 1995; O'Reilly et al 2014

This distinction between cortex and hippocampus motivates a classic account, known as complementary learning systems (CLS) theory. This theory argues how the hippocampus and cortex have evolved to work together to prevent catastrophic interference in biological systems. The theory argues that the cortex plays a role similar to that of a standard neural network, in that it learns slowly and incrementally from new sensory data, allowing a general sense of "familiarity" with objects or categories to be obtained from diverse experience. The hippocampus and other MTL structures, by contrast, learn very rapidly from new information, acquiring new episodic memories in a "one-shot" fashion, or after a single exposure, and supporting a distinct class of memory storage that supports recollection.

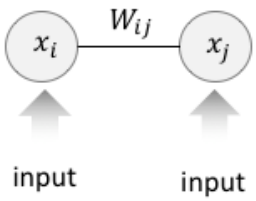
According to CLS, episodic memories are stored in a way that allows them to be recollected or "replayed" at a later time, even during periods when other learning is taking place, so that they too have the opportunity to be encoded slowly in the cortex. Thus, the recollection of episodic experiences provides an offline "interleaving" whereby old memories and new experiences are jointly encoded in the neocortex. This allows the network to find a minimum that simultaneously satisfies the demands of current and past tasks.

CLS involves 4 key ingredients:

1. Autoassociation via Hebbian learning
   - Acts as a parametric storage mechanism
2. Pattern separation
   - ensures hippocampal memories are distinctive and non-overlapping by sparsification
3. Experience-dependent replay
   - allows memories to be interleaved with ongoing experience, decorrelating inputs to the cortex
4. Consolidation
   - Memories are gradually folded into knowledge in the neocortex

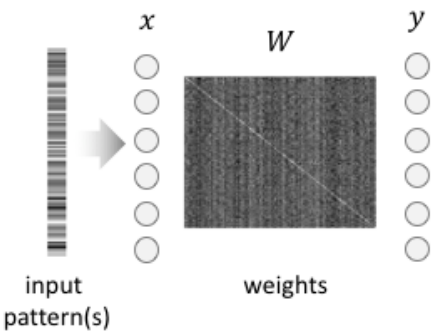
The complementary learning systems account is an important theoretical perspective and so we will take some time to unpack it in detail. Broadly, in contemporary versions of the theory, there are 4 major components: Hebbian autoencoding, pattern separation, experience-dependent replay, and neocortical consolidation.

$$W_{ij} = W_{ij} + (x_i \cdot x_j) \cdot \alpha$$



The connection strength between two neurons (in the same or different layers) is increased when they receive contiguous input, i.e.
"neurons that fire together, wire together"

Provides a **parametric** storage mechanism



Training:

$$W = W + (x \cdot x^T) \cdot \alpha$$

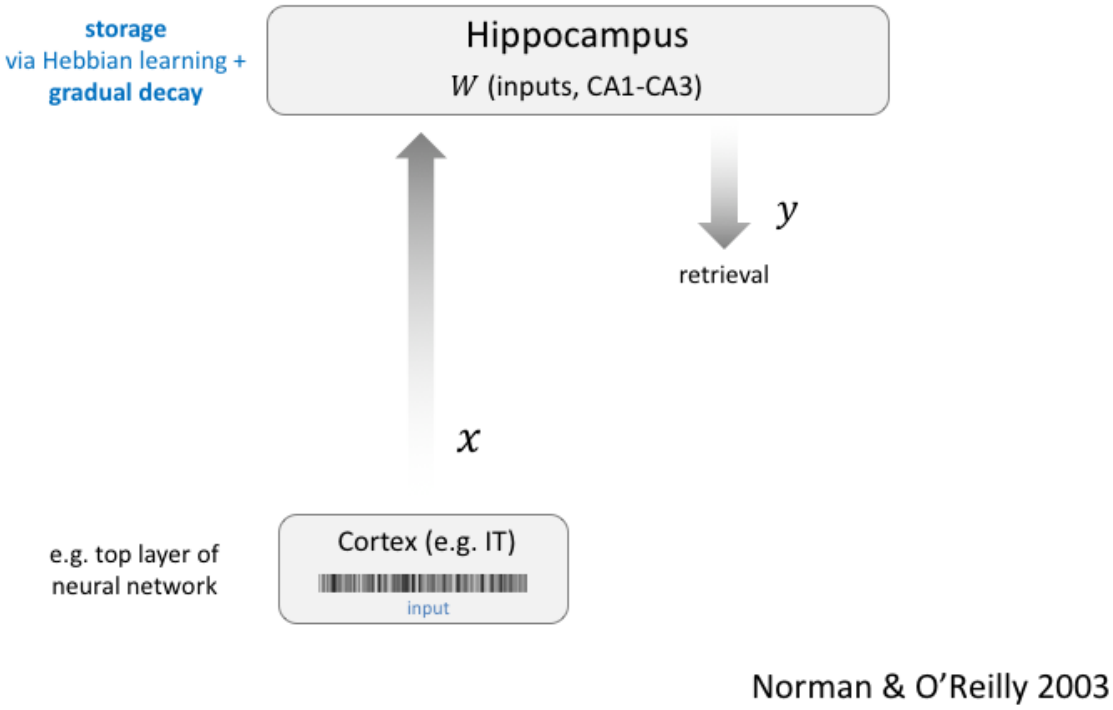
Test on example k:

$$y_k = x'_k \cdot W$$

y will be equal (or very similar) to x even if x' is a degraded version of x (fault tolerance)

We will consider unsupervised methods in much more detail in the next lecture, but for our current purposes, it suffices to understand how information can be stored in a network in a fault-tolerant fashion according to Hebbian principles. Recall the basis for Hebbian learning: neurons that are activated by a common input have their mutual connections strengthened, or as it is often said, "neurons that fire together wire together".

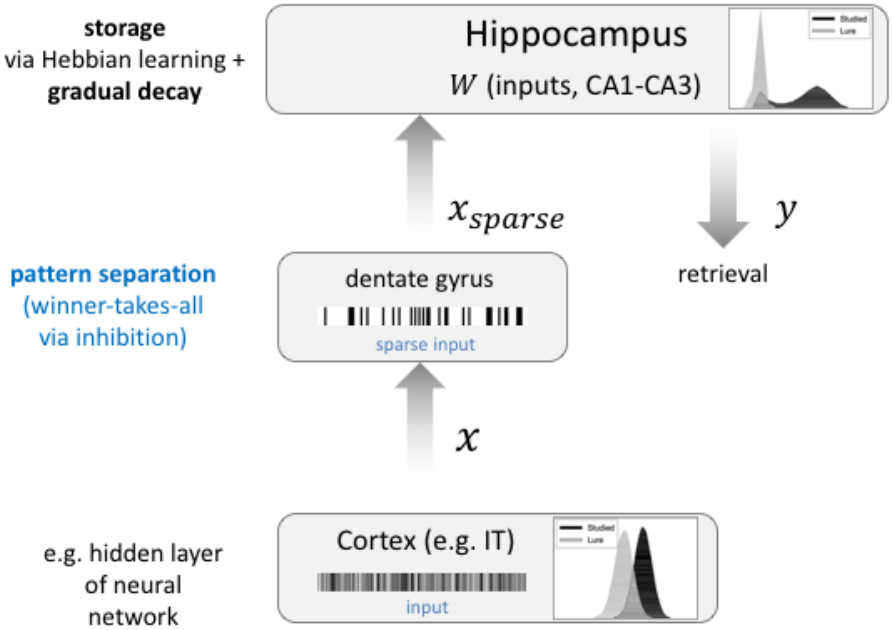
To understand that basic principle of autoassociation[65], consider a population of n neurons that are fully connected to each other (and, for convenience, themselves) by a weight matrix W that will be of size $n \times n$. When a common input arrives to neurons x_i and x_j these neurons become more strongly connected through increased synaptic strength. This can be implemented by updating weight W_{ij} in proportion to $x_i \cdot x_j$. More generally, for any input x we can train $W = W + (x \cdot x^T) \cdot \alpha$ where α is the learning rate, and $x \cdot x^T$ is the outer product of x, i.e. itself multiplied by its transpose (hence "autoassociation"). Critically, a network of this form can be trained to encode multiple distinct inputs in such a form that when probed with a degraded version of x (for example x') by multiplying that degraded input by the weights $x' \cdot W$, then a vector is elicited that will reproduce the original input with a reasonable level of fault tolerance. An autoassociative network of this form can thus be thought of as a parametric storage mechanism, i.e. an encoding model where the number of units (i.e. neurons) does not have to grow with each new memory that is formed.



Norman & O'Reilly 2003

[65] This book is generally a bit technical but the chapter on autoassociation is relatively clear: https://page.mi.fu-berlin.de/rojas/neural/chapter/K12.pdf

So broadly, we can think of the MTL circuitry (including the hippocampus) as receiving input from a cortical neural network (for example, the inputs x could be from a hidden layer of that network) and acting as an autoassociative storage mechanism. Random activity passing through the network at a later date will allow these inputs to be retrieved.

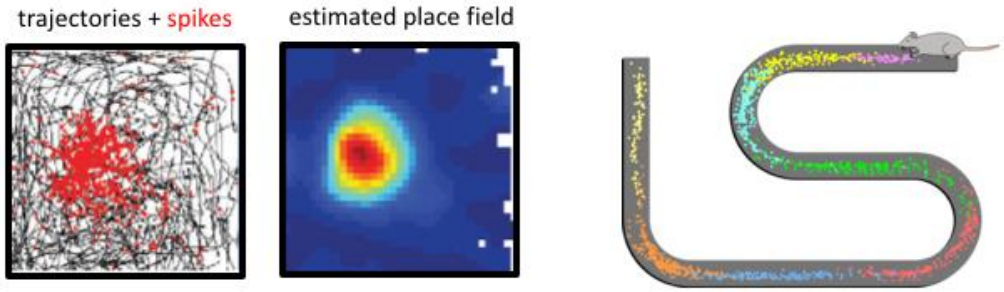


Norman & O'Reilly 2003

As we shall see later in the course, the storage capacity of autoencoding methods (including the Hebbian autoassociation described here) can be increased when the inputs are sparse. This means, for example, that inputs to the storage system undergo a preprocessing step in which the stronger inputs are strengthened and the weaker inputs are weakened. This reduces the correlation among inputs and makes them more distinctive. There is good evidence that the dentate gyrus (DG), through which inputs pass *en route* from the cortex to the hippocampus, performs a computational set that resembles *pattern separation*, making the sensory inputs sparser and less overlapping[66]. This might happen for example via a winner-takes-all inhibition mechanism that places inputs in mutual competition and maintains only the activity in the most active units. Recordings from the DG seem to be particularly sparse – in other words, only a very small fraction of neurons is activated by a single synapse. In general, the sparsity of MTL units (as described in classic papers showing the high degree of neural selectivity in this region) may be a computational step which helps increase storage capacity.

6.2. Experience-dependent replay and consolidation

[66] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2976779/

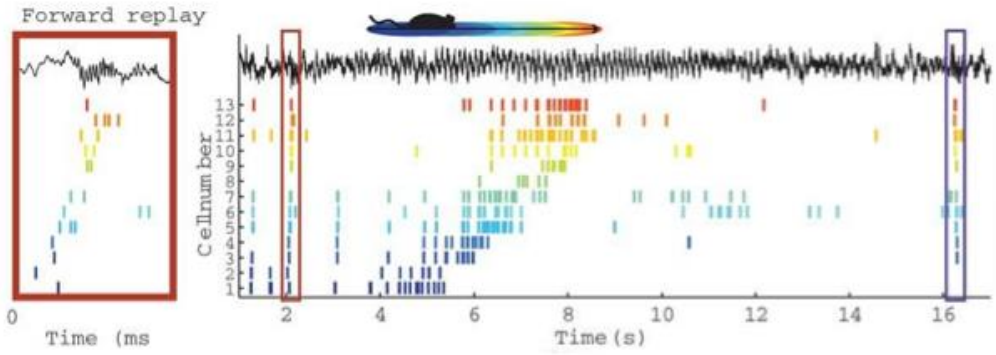


Place cells are sparse representations, coding for a unique locations in space as a rodent explores an open arena or runs on a track

O'Keefe & Nadel 1976

An example of the sort of representation that is observed in the hippocampus (at least in rodents) is a place cell. Place cells fire when the animal occupies a specific location in an environment such as a testing box or running track, but are less sensitive to the head direction or other corollary sensory information[67].

[67] Hartley T, Lever C, Burgess N, O'Keefe J. Space in the brain: how the hippocampal formation supports spatial cognition. Philos Trans R Soc Lond B Biol Sci. 2013 Dec 23;369(1635):20120510.



During inactivity (and sleep), sharp-wave ripples reinstate patterns of place cell activity recorded during earlier locomotion, as if the animal were "replaying" or "reimagining" experienced events (20x faster)

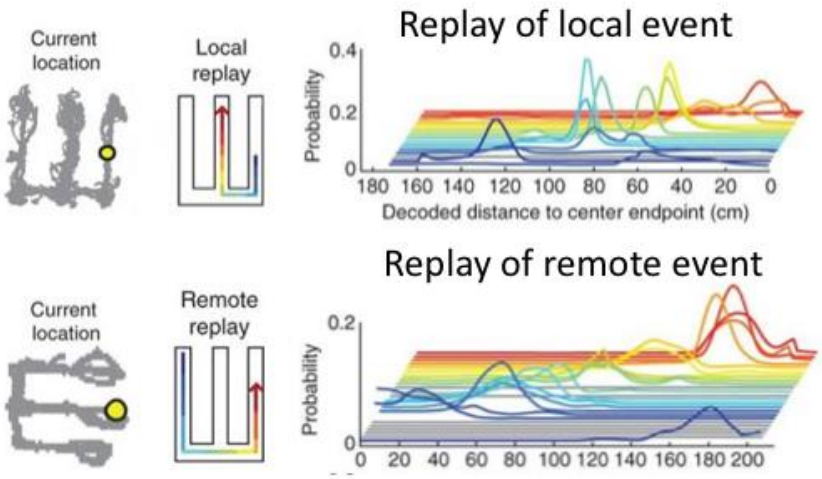
These signals propagate to the cortex

Carr et al 2014; O'Neill et al 2010

The key proposal of CLS is that information retrieved from hippocampal storage system can be "replayed" in a fashion that interleaves past experience with ongoing sensory inputs[68], to allow the network to break the temporal autocorrelation that is inevitable in natural environments. There is now excellent evidence for this sort of "replay" mechanism in the hippocampus (in fact, replay also occurs elsewhere in the brain, such as the PFC, but we will focus on the hippocampus here). During sleep or quiet resting, hippocampal cells exhibit fast bursts of activity known as "sharp wave ripples", during which place cells become rapidly active in sequence. Most interestingly, the sequence of activation tends to restate that which was experienced during recent activity, only an order of magnitude faster. Thus, if an animal runs repeatedly on a linear track whose locations are encoded in place cells numbered 1,2,3..n then during these "replay" events the cells will reactivate in that specific order, even when the rat has been removed from the track. In other words, it is as if the animal is "replaying" or "reimagining" past experiences in a structured fashion, allowing the interleaving of past experiences with current sensory data. CLS argues that this mechanism is critical for reducing catastrophic interference[69].

[68] Lots of good reviews on this topic. Ólafsdóttir HF, Bush D, Barry C. The Role of Hippocampal Replay in Memory and Planning. Curr Biol. 2018 Jan 8;28(1):R37-R50. doi: 10.1016/j.cub.2017.10.073. Foster DJ. Replay Comes of Age. Annu Rev Neurosci. 2017 Jul 25;40:581-602. Carr MF, Jadhav SP, Frank LM. Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. Carr MF, Jadhav SP, Frank LM.

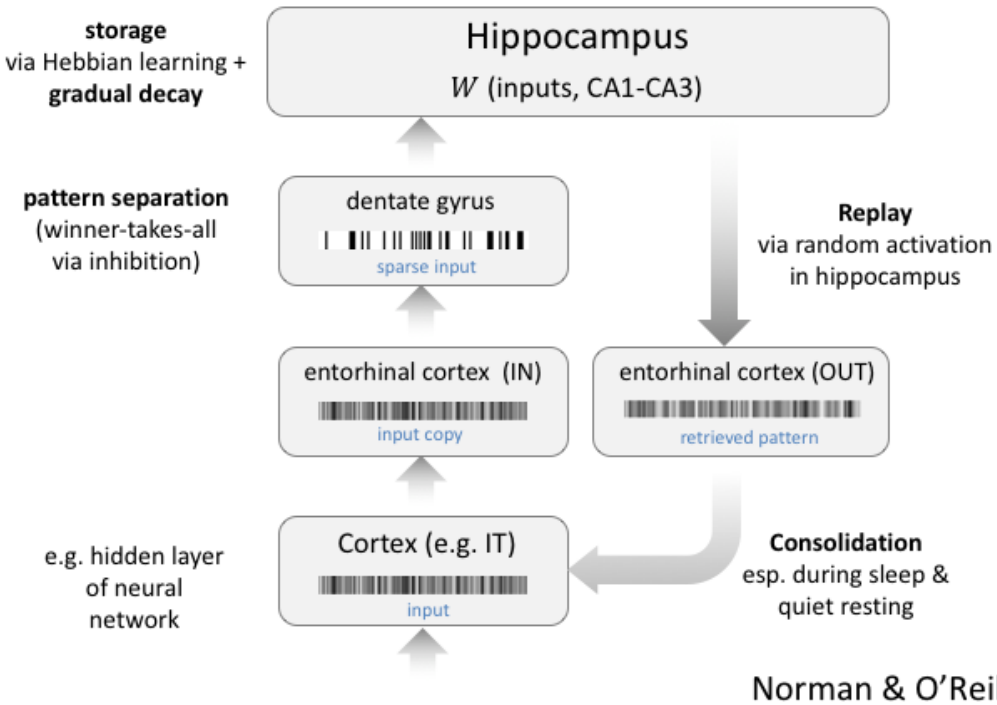
[69] Kumaran D, Hassabis D, McClelland JL.
What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. Trends Cogn Sci. 2016 Jul;20(7):512-534.



In the active state, local and remote events are replayed, allowing for interleaving of ongoing and remote experience

Carr et al 2014; O'Neill et al 2010

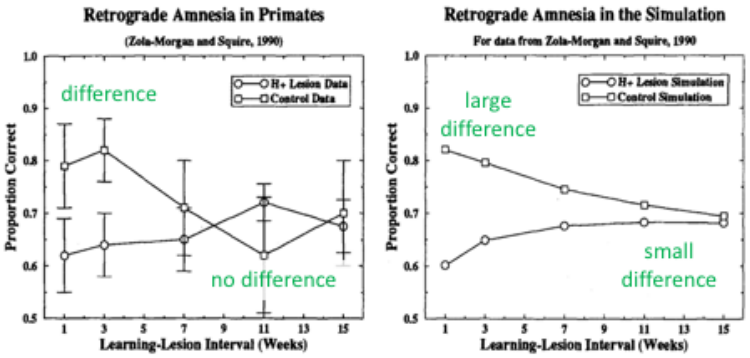
Replay can occur during ongoing behaviour, or at a subsequent time when the animal has been removed from the testing environment. Whist it is in the testing environment, the replay can be either local or remote; in other words, the animal can use replay to explore the transition structure of the environment, and to learn more about previously experienced trajectories, in order to facilitate future behaviour. In fact, replay may have a role which goes beyond the simple interleaving of past and future experience, as we shall see below.



Norman & O'Reilly 2003

Information that is retrieved during replay is passed back to the cortex via the entrorhinal cortex, allowing memories to consolidated in cortical circuits. This completes the cortico-hippocampal-neocortical loop, so that information encoded in the cortex contains a mixture of experiences.

Hippocampal lesions provoke retrograde amnesia for recent events, as well as anterograde amnesia



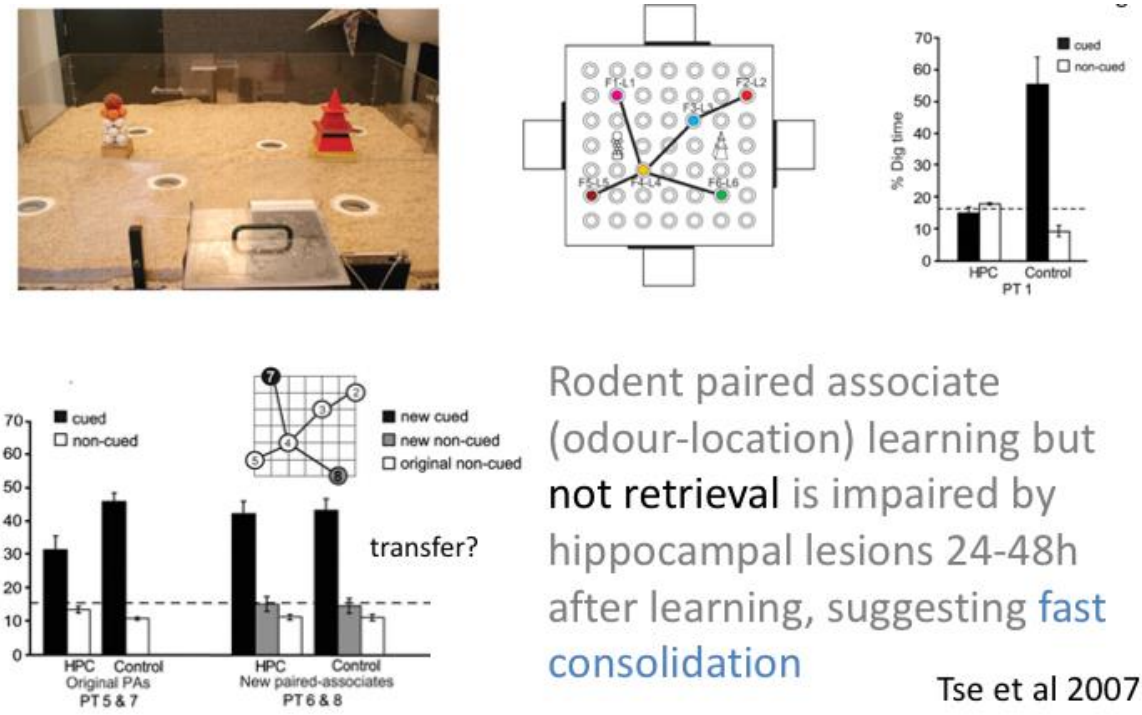
Hippocampal-lesioned monkeys show selective retrieval impairment for information learned proximal to the lesion

This effect is recreated by curtailing reinstatement (at lesion) in the neural network model

Zola-Morgan & Squire 1990

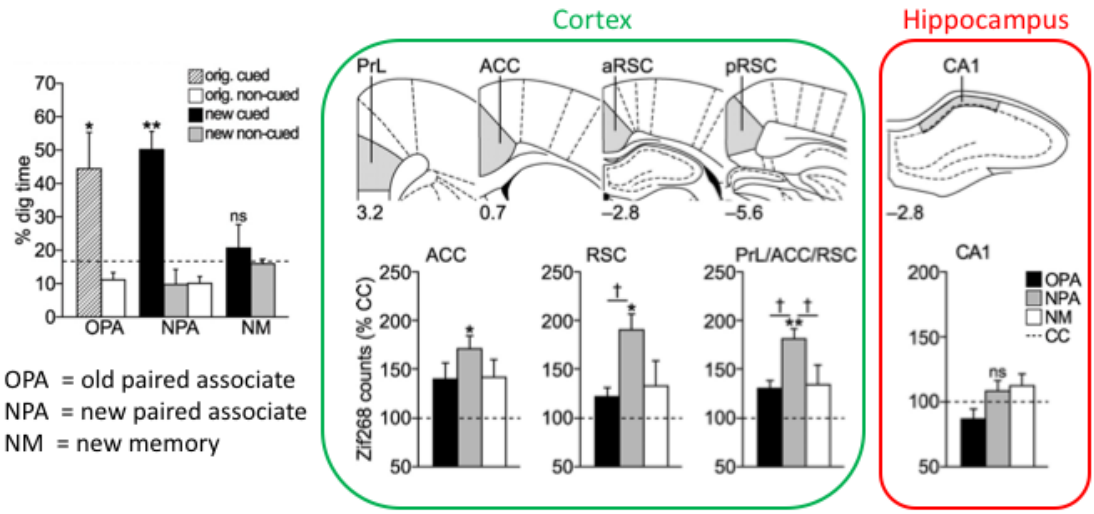
CLS thus explains why hippocampal lesions lead both to anterograde amnesia (because new information cannot be encoded in a one-shot fashion) and a gradient of retrograde amnesia, whereby information that was learned immediately prior to the lesion tends to be most vulnerable to hippocampal damage. For example, in this study by Zola-Morgan & Squire[70], relative to control monkeys there was no difference in retrieval of information that was learned >7 weeks prior to a hippocampal lesion, whereas information that was learned just a few weeks previously was impaired. According to CLS, this is because the longer-lag information had been consolidated into cortical circuits, where it was protected from interference by the lesion. A neural network model that involved a consolidation step was able to recreate this pattern of data.

[70] Zola-Morgan SM, Squire LR. The primate hippocampal formation: evidence for a time-limited role in memory storage. Science. 1990 Oct 12;250(4978):288-90. See 1995 CLS paper for neural network comparison.



However, it is worth noting that CLS predicts that consolidation should be slow, occurring as information is gradually replayed in the hippocampus and re-encoded in the neocortex. However, there is also evidence that consolidation can occur remarkably rapidly. In this classic task by Tse and colleagues[71], rats learned to forage for food in an open area. Food locations were cued by an odour signal that was given at the start of the trial. If a hippocampal lesion was made at before training occurred, then animals could not acquire the paired associations (top right). Subsequently, however the experimenters trained the animals normally, and then introduced a single trial of a new paired association, in which two new odours were paired with locations near to previously trained food wells. The animals were able to learn this association in a single shot, as demonstrated by an unrewarded probe 24h later. Critically, however, this memory survived a subsequent hippocampal lesion. This suggests that even if the hippocampus is required for paired associate acquisition, it can be consolidated to the neocortex extremely rapidly – even after a single learning event. This presents a challenge to the view proposed by CLS, which argues that hippocampal learning is fast and neocortical consolidation is much slower.

[71] Schemas and memory consolidation. Tse D, Langston RF, Kakeyama M, Bethus I, Spooner PA, Wood ER, Witter MP, Morris RG. Science. 2007 Apr 6;316(5821):76-82.



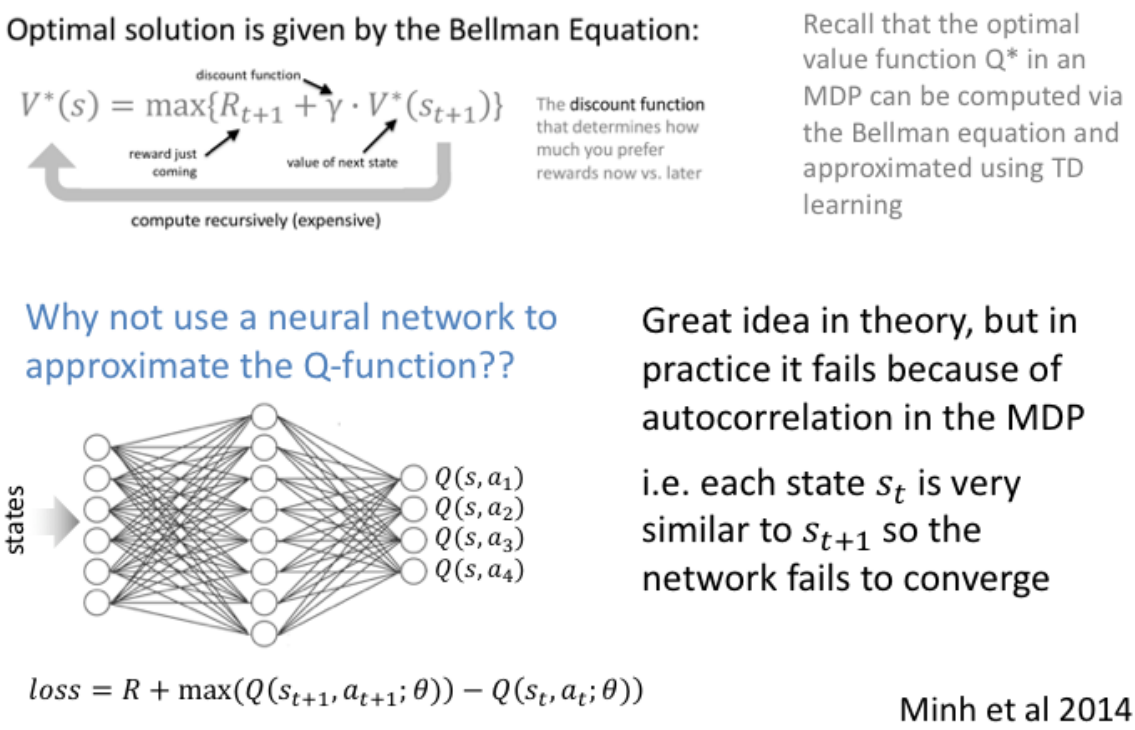
New paired-associate learning associated with early immediate gene expression in cortex but not hippocampus immediately after learning

Tse et al 2011

In a follow-up study, the same authors[72] measured early immediate gene expression – a measure of protein expression that is linked to synapse formation – in the hippocampus and cortex during the same paradigm. They found that the new paired-associate learning was linked to gene expression in the cortex (including prelimbic regions of the medial PFC) but not the CA1 region of the hippocampus, providing support for the view that the effect is mediated by cortical learning mechanisms. So, a complete biological account of knowledge acquisition needs to be able to explain how information can be consolidated rapidly to the neocortex, potentially requiring additional mechanisms to those proposed by CLS.

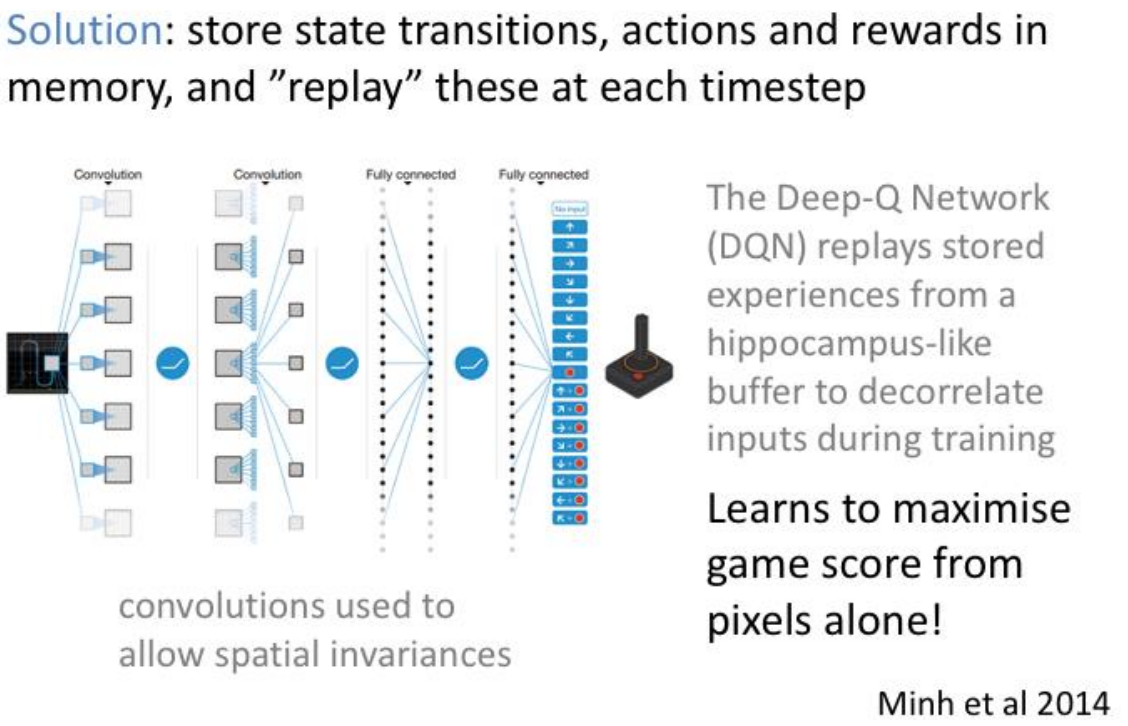
6.4. Function approximation for RL: the Deep Q-network

[72] Tse D, Takeuchi T, Kakeyama M, Kajii Y, Okuno H, Tohyama C, Bito H, Morris RG. Schema-dependent gene activation and memory encoding in neocortex. Science. 2011 Aug 12;333(6044):891-5.



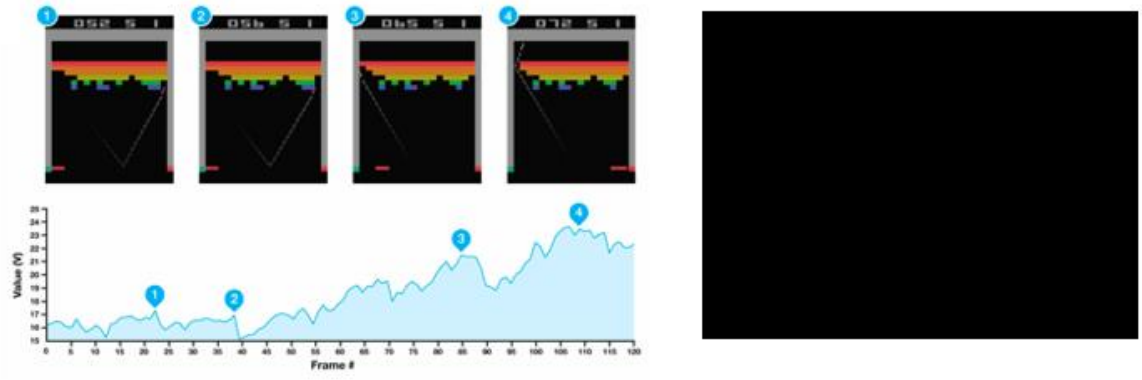
CLS is an interesting theory of how learning occurs in the neocortex and hippocampus of biological agents. But can it be practically applied to improve performance in machine learning and AI research? The answer is yes, as we shall see. But to introduce how, we need to return to lecture 2, where we discussed model-free RL methods that allowed an agent to learn, in tabular form, the optimal value of states/actions in an MDP, as given by the Bellman equation. One of the limitations of the type of approach discussed is that the models were "nonparametric", in that the model size is obliged to grow as the number of states/action pairs in the world. Ideally, we would like to use a parametric model to learn by reinforcement. How can we do that?

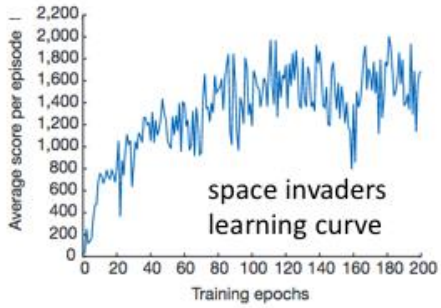
Well, in principle there is nothing to stop you from using a functional approximator, such as a neural network, to learn the optimal Q values in an RL task. All you would need to do is optimise the network to minimise the prediction error associated with each state; when the prediction error is zero, the network has converged and learned the optimal Q-values. This sounds simple, but in fact it's very tricky – because typically the sorts of environments where one might wish to use an RL model (e.g. the grid world discussed in lecture 2) exhibit strong patterns of temporal autocorrelation – in other words the state s_t is physically very similar to the state s_{t+1}, creating a situation similar to the "blocking" of tasks that provokes catastrophic interference in the supervised setting.



However, replay offers a potential solution to this problem, embodied in the "Deep Q Network" (DQN). Described in a paper in 2015[73], DQN successfully learned to play more than 30 Atari 2600 video games at near- or super-human levels. Architecturally, DQN is a deep convolutional neural network that learns the optimal Q function as described above but avoids instability by storing and periodically replaying memories of past events alongside new inputs. This allows it to break the correlation between successive video frames in the Atari environment and to learn a function that maps pixel inputs onto predicted rewards. However, DQN's storage system is far more primitive than the mammalian hippocampus – it simply saves the past million video frames into a large memory buffer, without any compression or other preprocessing.

[73] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature. 2015 Feb 26;518(7540):529-33.

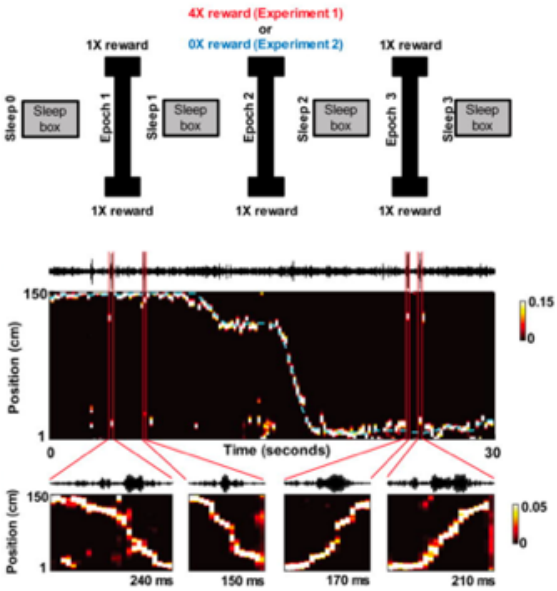


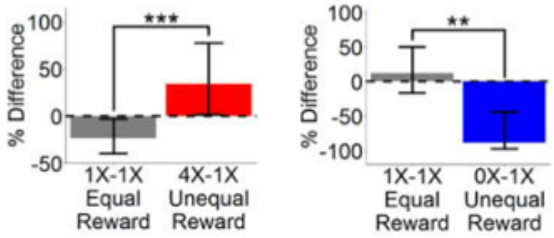


The network performs at > 75% of a human expert on 29/49 games

Minh et al 2014

An example of the network playing space invaders is shown here. The network begins with chance-level play (as the weights are initialised to random) but gradually learns to perform as an expert. In the game breakout (top left), you can see how the state value function grows as the network gradually "tunnels" through the wall towards the score-maximising goal of trapping the ball above the bricks.



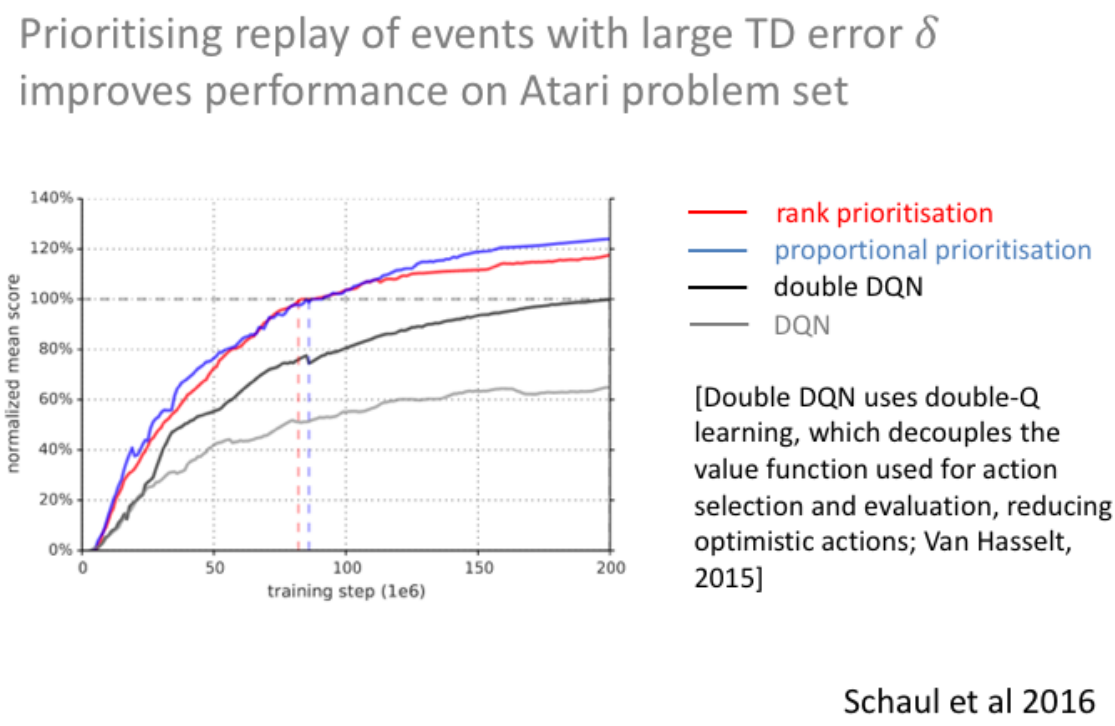


Replay events occurred more frequently towards the 4X rewarded end in Exp1, and less frequently towards the 0X (unrewarded) end in Exp2

replay events recorded during maze foraging task

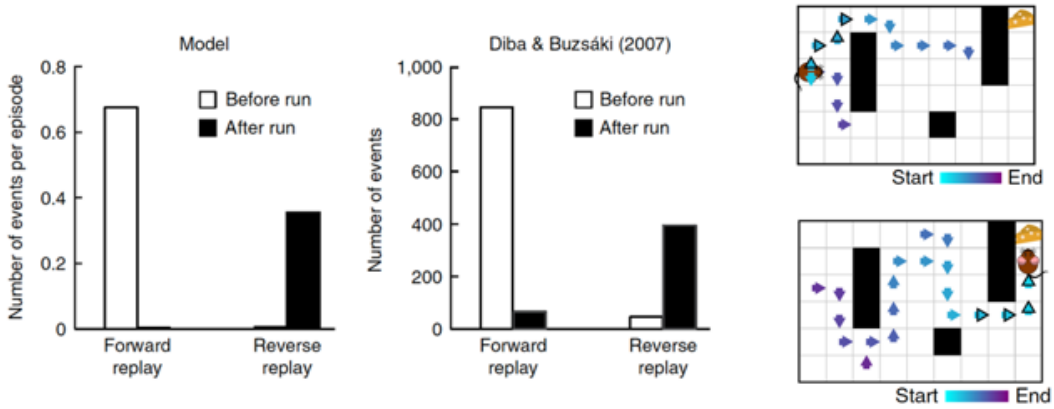
Ambrose et al 2016

DQN replayed events at random, which maximally decorrelated the past and current input to circumvent the problem of catastrophic interference. However, this may not be the most efficient strategy. Subjectively, our internal rumination tends to dwell on events that are important or salient for future behaviour, such as those that incurred strong positive or negative outcomes (PTSD is an example of a disorder in which replay of a salient negative event occurs so frequently that it becomes disruptive). Indeed, in rodents, replay events tend to happen preferentially near highly rewarded events, as shown in this paper by Ambrose and colleagues[74], in which the magnitude of available rewards in a maze was carefully controlled. Replay events occurred more frequently at the location of a large reward, and less frequently at the location of an absent reward, relative to the control condition (small reward).



In fact, incorporating a similar "prioritised" replay scheme into a variant of DQN[75], such that events leading to large prediction errors were more likely to be replayed, substantially increased performance relative to DQN versions using only random replay.

[74] Ambrose RE, Pfeiffer BE, Foster DJ. Reverse Replay of Hippocampal Place Cells Is Uniquely Modulated by Changing Reward. Neuron. 2016 Sep 7;91(5):1124-1136.
[75] https://arxiv.org/abs/1511.05952



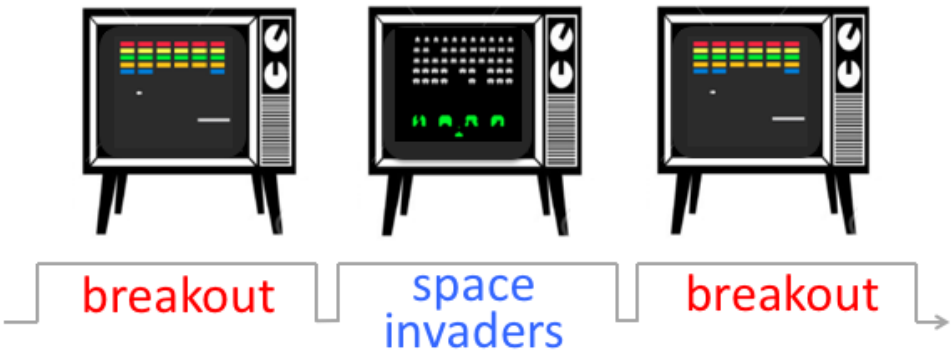
An optimal model that derives the reward-maximizing replay policy explains why reverse replay is observed at goal locations, but forward replay prevails far from goal

Mattar et al 2018

In fact, a recent paper[76] offers a more complete account of the constraints under which replay should occur to facilitate learning, under the assumption that learning is largely model-free (but that simulation can occur offline). The authors used simulation to identify which patterns of replay would maximise the rate of reward learning in simple navigational environments, such as linear tracks and mazes in which replay has been most intensively studied. Their work shows that many salient features of empirically observed replay in rodents are in fact reward-maximising strategies. For example, replay can occur forwards (from the current location to future states) or backwards (from the current location back to past states). The authors' optimal model suggests that backwards replay is optimal when a reward has just been experienced (because it backs up the reward to recent states) whereas forward replay is optimal when the goal is more distal (because it maximises the chances of replaying a potentially fruitful route to the goal). This is exactly what the is observed in empirical studies, for example one by Diba & Buszaki (2007) as shown on the slide.

6.5. Knowledge partitioning and resource allocation problem

[76] Mattar MG, Daw ND. Prioritized memory access explains planning and hippocampal replay. Nat Neurosci. 2018 Nov;21(11):1609-1617.

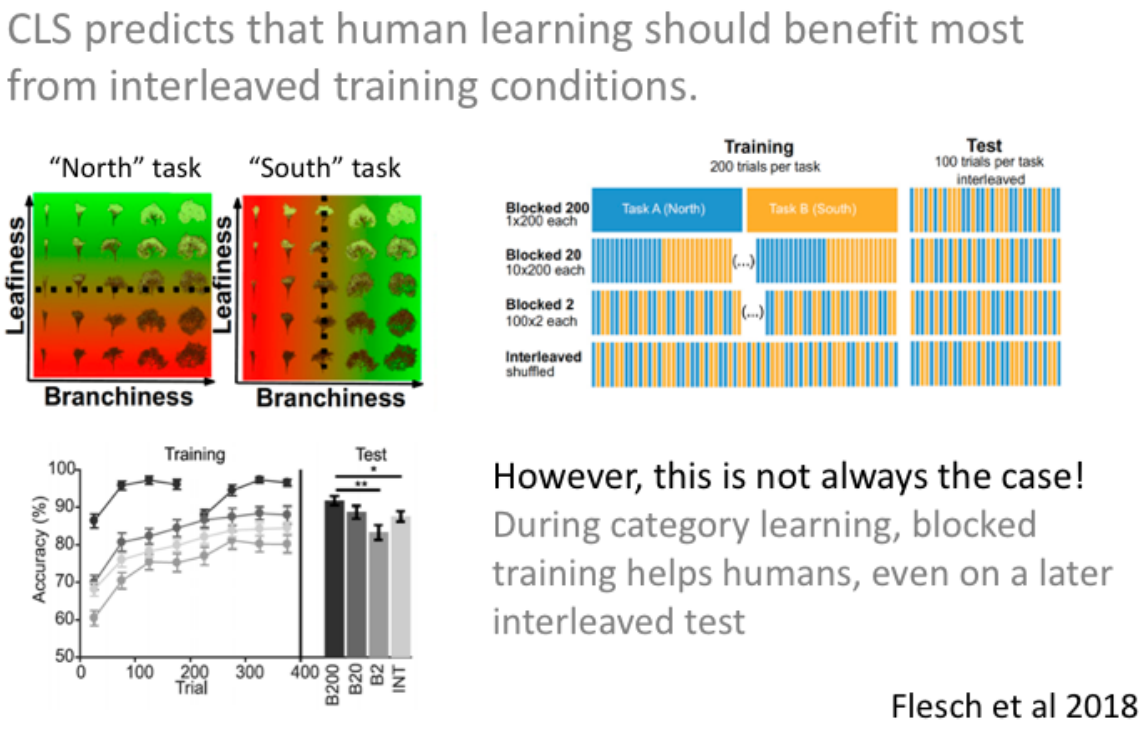


Continual learning remains an unsolved problem

Atari success is only possible if network weights are reinitialised to random between games

Mnih et al 2014

Despite these advances, and despite the evident success of replay both as a strategy for RL in dynamic environments and as an explanation for neural data recorded in rodent experiments, continual learning remains an unsolved problem in both machine learning and neuroscience. For example, in the Atari environment, DQN was able to learn to play multiple games at superhuman levels, but only if its memory was "reset" (i.e. the weights were randomly reinitialised between games). Of course, this is very different from human play – where the same expert can potentially learn to play all of the games using the same model (i.e. brain). Until we know how to build brains that can continue to learn in a way that avoids mutual interference between past and current knowledge, we won't be able to build strong AI.

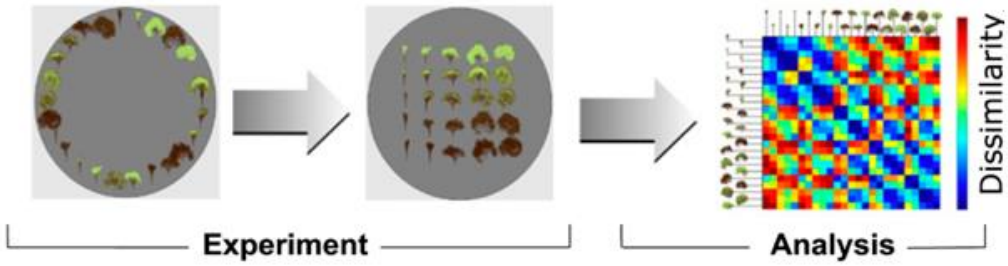


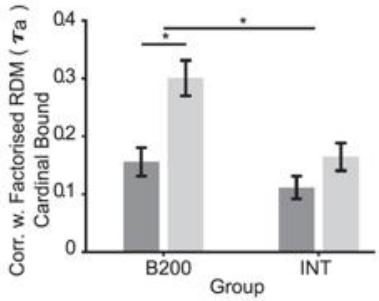
Another important caveat is that CLS assumes that human learning benefits most from interleaving. This is clearly true in some cases. There are examples from the domains of skill learning (e.g. sports), language learning and even more abstract capacities (e.g. mathematical ability) that support this notion (for example, a large memory literature has emphasised the benefits of spaced over massed practice). However, it's not the case that full interleaving is always beneficial. Think about the challenge of learning both French and Spanish – you probably don't want to mix up vocabulary learning from the two domains in the same lesson!

The benefits of blocked rather than interleaved training were demonstrated in this recent study by Flesch and colleagues[77]. They asked participants to classify a space of naturalistic stimuli (trees) into one of two categories on the basis of trial-and-error feedback alone. Critically, the classification rule (plant the tree according to its leafiness vs branchiness) was varied in two different contexts. Participants in different groups experienced these contexts either in long blocks or randomly intermixed, but during testing (without feedback) all contexts were interleaved. Those who had experienced blocked training performed better at interleaved testing, even better than those who had experienced exactly the same interleaving during training.

The authors then examined the pattern of errors that humans made during the task. They found that the benefits of blocked training over interleaved was particularly due to participants learning the two orthogonal task boundaries – as if blocked training helped participants "factorise" the problem into two distinct tasks. So for humans, in contrast to neural networks, learning to segregate information according to context may be beneficial.

[77] Flesch T, Balaguer J, Dekker R, Nili H, Summerfield C. Comparing continual task learning in minds and machines. Proc Natl Acad Sci U S A. 2018 Oct 30;115(44):E10313-E10322.

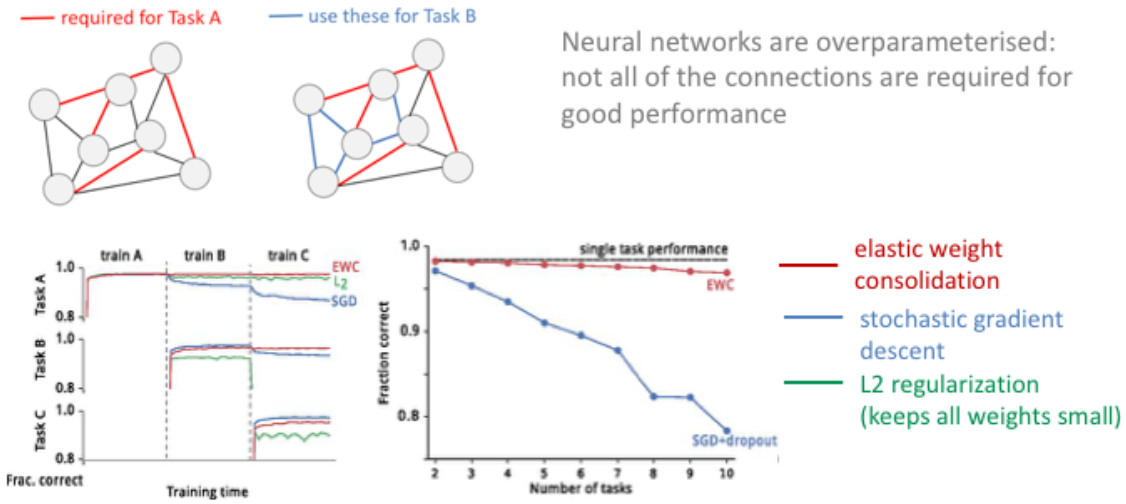




Human prior beliefs about the structure of tree space predicted the benefit of blocked training, as if **factorising** the overall problem into two distinct tasks was helpful

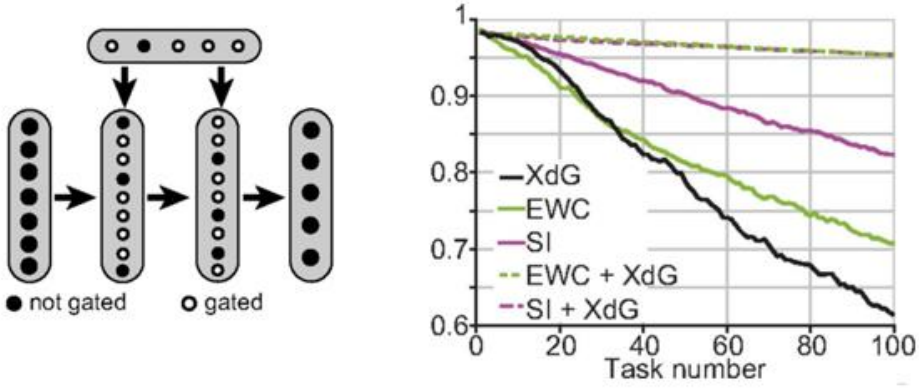
Flesch et al 2018

This was supported by a further experiment in which the researchers measured participant's prior understanding of the "tree space" – the way they represented the naturalistic stimuli that were generated for the experiment. Those participants that naturally represented the trees as being organised according to orthogonal "leafiness" and "branchiness" displayed the most benefit from blocked training, as if this training regime reinforced the partitioning of knowledge according to context.



Other approaches from machine learning attempt to allocate task information to distinct synapses, using a technique known as "stabilisation". In this paper by Kirkpatrick and colleagues[78], capitalising on overparameterisation, they identify those network weights that are most important for a given task, and "freeze" them selectively by slowing down the learning rate. This allows the new learning to be allocated in a way that makes it less likely to interfere with old knowledge, conferring a substantial protection against catastrophic interference both in toy domains and in Atari games. They call this "elastic weight consolidation". A related method, known as "synaptic intelligence" performs similarly[79].

[78] Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R. Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci U S A. 2017 Mar 28;114(13):3521-3526.
[79] https://arxiv.org/abs/1703.04200



Stabilisation approaches are particularly effective in conjunction with task-dependent gating, in which synapses are randomly allocated to tasks

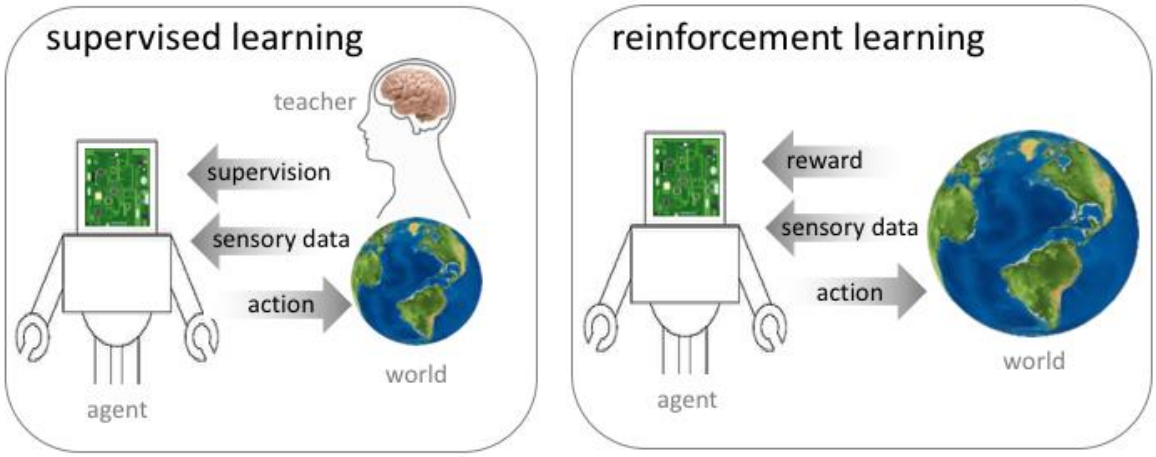
Masse et al 2018

It was recently shown[80] that these techniques are particularly effective when combined with another method, that randomly gates synapses for each context, ensuring more limited overlap between the weights that participate in each task. For example, this might be one function of the top-down signals that are observed during task-level control in monkey and human neural recordings. These methods, which ensure that knowledge is partitioned, are likely to be promising avenues for solving continual learning in the future.

[80] Masse NY, Grant GD, Freedman DJ. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. Proc Natl Acad Sci U S A. 2018 Oct 30;115(44):E10467-E10475.

7. Unsupervised and generative models

7.1. Unsupervised learning: knowing that a thing is a thing



The ML methods we have considered so far require dense feedback signals from the environment

In the real world, these signals are not always available

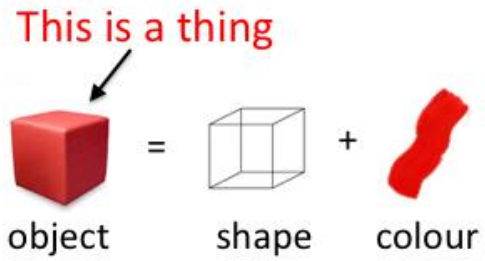
Thus far in the course, we have considered two canonical machine learning methods: model-free reinforcement learning and supervised learning. Both of these frameworks assume that information is sent from the world to the agent in two distinct forms: as observations (e.g. sensory signals) and as feedback (e.g. reward or teaching signals). We have seen that model-free RL methods are limited when the reinforcement provided by the environment is sparse. This is often the case: for example, humans engage in complex patterns of behaviour in order to achieve a distant goal, without necessarily receiving interim rewards along the way. Supervised learning requires an "oracle" or teacher to provide information about the correct answer after every decision. In natural environments, this teaching signal may not always be available.

The developmental literature demonstrates that infants learn about objects, words and their interactions in the first year of life, i.e. before they can produce complex motor behaviours or language

This knowledge is acquired without direct supervision or reinforcement

One explanation is offered by nativist theories, which argue that this knowledge is inborn. This is clearly true in part, but there is another mode of learning that is very important…

Humans and other animals, thus, can learn even where feedback is limited or absent. Developmental psychology illustrates this point in great detail. In the first year of life, when most infants understand only a very limited vocabulary, they nevertheless learn a great deal about the world, for example, being able to distinguish between various classes of object. There must be a mechanism, thus, by which knowledge is acquired with minimal supervision or reinforcement. Whilst it is true that a predisposition towards certain forms of knowledge (e.g fear of spiders) may be provided by genetic heritage – as proposed by nativist theories – we clearly need another mechanism to account for the power of biological learning.



Young infants understand that objects are distinct from the background, can be occluded or hidden, are solid, have immutable properties like shape and colour etc.

Object understanding is more than just assigning a category label. **Infants must learn that an object is a "thing"**

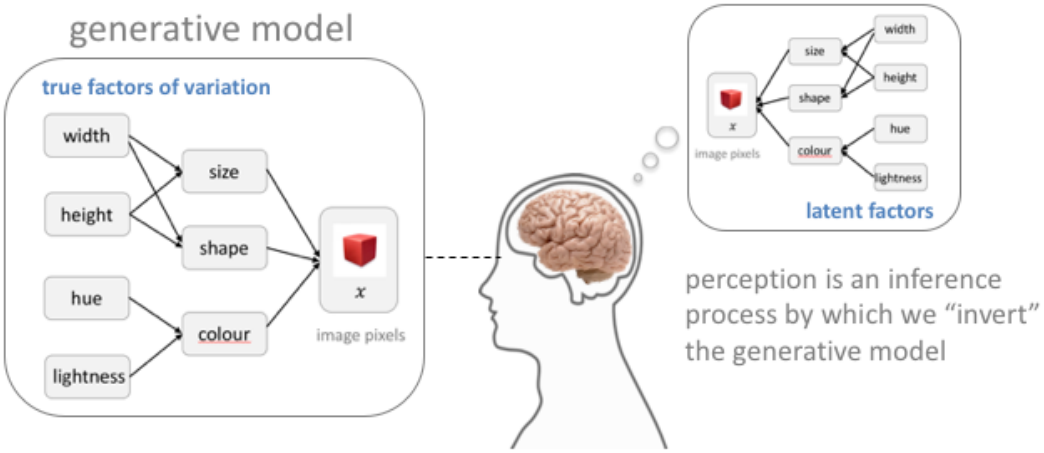
For example, one important source of knowledge that is acquired without feedback or reinforcement is about the nature of object. Infants learn from an early age that objects are distinct from one another, for example that an object might still be present even when partly occluded, that objects are solid and cannot occupy the same space, etc (Elizabeth Spelke's work is instrumental here[81]). The knowledge that infants acquire about objects goes way beyond simply assigning a category label: they understand how objects are composed of features, how they relate to one another, and how they are subject to basic laws of physics. How do they acquire this knowledge?

[81] Too many papers to read here, but take a look: Spelke's work is among the most important in the field https://www.harvardlds.org/our-labs/spelke-labspelke-lab-members/elizabeth-spelke/

The success of adversarial networks suggests that deep supervised neural networks have very limited object understanding

They just learn a mapping from image statistics to labels, and do not infer the true structure of the world

Recall that in lecture 3 we discussed adversarial neworks, that laid bare the limited object understanding displayed by deep neural networks. It's easy to fool deep networks – although they may learn to classify objects accurately on average, it's possible to identify situations where they will make nonsensical categorisation decisions. Deep neural networks learn how a large conditional distribution over image pixels maps onto a class label, but they don't really understand what objects are and how they behave. In part, this might be because neural networks typically only receive object information through a single sensory channel (vision), in contrast to human infants, who can handle and interact with objects using their behaviour. However, there is another machine learning method that offers more promise for teaching machines to have greater object understanding.

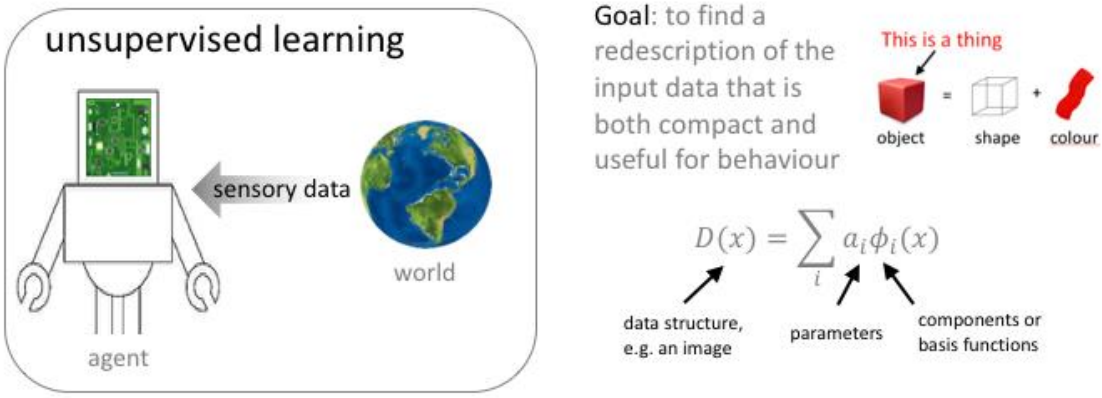


One way of thinking about knowledge as encoding a **model of the world**, that is an inversion of the true data-generating process that gives rise to (noisy, ambiguous) sensory inputs

To understand unsupervised methods, we have to take a step back and consider a whole different framework for thinking about sensory data and the nature of reality. A long philosophical tradition – dating back to the Greeks – sees perception as an "inference" process that attempts to reconstruct the true structure of the world from limited and noisy data. In psychology, this tradition can be traced back to Helmholtz's notion of "unconscious inference", and subsequently through the influence of the constructivist movement in visual perception, most often associated with figures such as Richard Gregory[82].

This tradition considers the world as being best described as a set of unobservable processes (or "latent variables") that give rise to sensory data. The information impinging on our senses is thus a partial or noisy reflection of these processes, and the task of perception is to reconstruct the true nature of the world from this data. We can thus think of the world as being constituted by a "generative model" that generates sensory data, for example the patterns of light incident on the retina. Perceptual processes have evolved to "invert" that generative model, to infer the true causes of sensation.

[82] Gregory's book Eye and Brain is the classic text here.



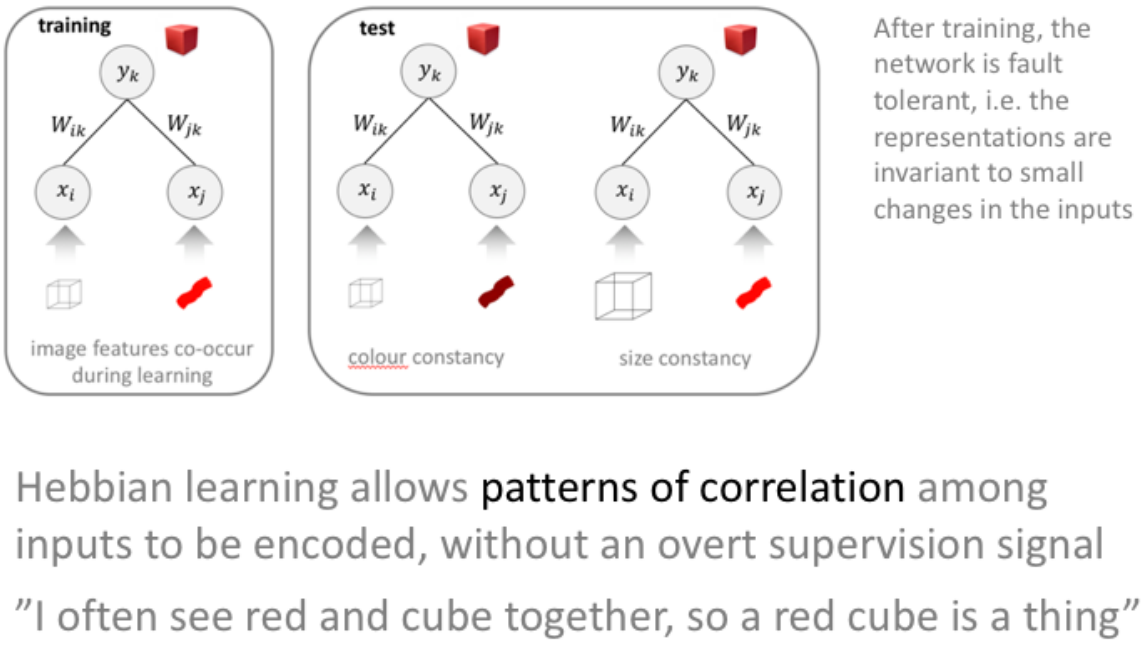
In unsupervised learning, the agent encodes and organizes knowledge about the world purely by receiving sensory inputs

Unsupervised learning methods are a class of machine learning technique that attempt to redescribe sensory data in terms of a (smaller) set of latent variables. Unsupervised methods are thus intrinsically related to the notion of dimensionality reduction, i.e. that we can take a high-dimensional input (such as image pixels depicting an object) and describe it compactly in terms of a set of variables (e.g. the shape, colour or size of the object). Critically, unsupervised methods do not make the (potentially artificial) distinction between sensory observations and feedback; all inputs are potentially indicative of the structure of the world, irrespective of whether they pertain to hedonic experience or not. This perspective allows us to circumvent the challenge of understanding what a "reward" is in the first place in natural environments (e.g. is the sight of a chocolate bar a reward? Its taste in the mouth? Or only the calorific benefit once it is ingested?). So unsupervised methods reformat sensory signals in useful ways, rather than mapping them onto an explicit label or reward that is given by the world.

The formula on the slide shows the canonical approach taken by unsupervised approaches. A data structure $D(x)$ is decomposed into a set of components $\phi(x)$ with each component ϕ_i being weighted by a parameter a_i. The components are the latent or generative processes that give rise to the data structure. Note that these are always inferred: there is no ground truth that says whether they are "right" or "wrong", and there are infinitely many ways in which the data can be decomposed. How, then do we know which is the best description of the data? Well, if our unsupervised model is a computational theory of the brain, then the best description is one that accurately captures the coding properties of sensory neurons. In machine learning, it's one that is able to accurately reconstruct sensory data from limited inputs using the lowest possible capacity. In other words, unsupervised methods are often optimised for efficiency. The idea that a neural information processing system should learn to

code as efficiently as possible for sensory data (e.g. using as few neurons as possible) is one that dates back to Barlow[83].

7.2. Encoding models: Hebbian learning and sparse coding



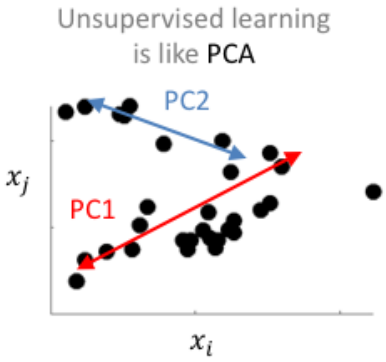
In lecture 5, we saw that Hebbian learning is a canonical unsupervised learning technique that can allow information to be stored efficiently, for example in an autoassociative network. Hebbian learning embodies the principle that statistically correlated inputs allow the formation of new composite representations. For example, let's imagine that inputs x_i and x_j correspond to the presence of two critical features of an input, such as its shape (square) and colour (red) If "square" and "red" repeatedly co-occur, then under Hebb's rule, their connections to a subsequent neuron will be jointly strengthened – as if the network has learned that red and square "go together", i.e. that in the dataset there is such a "thing" as a red square. As we have seen, Hebbian learning confers fault tolerance, so that even if one of the inputs is noisy or degraded, the input should be reconstructed accurately – for example, allowing for various forms of constancy.

[83] Barlow H.B. (1961). Possible principles underlying the transformations of sensory messages. Chapter 13. In: Sensory Communication, W.Rosenblith (Ed.), M.I.T. Press, pp. 217-234.

The goal is to learn a mapping of inputs x onto a target y via a function parameterized by θ

e.g. map images → labels

The simplest supervised Network (e.g. a Perceptron) directly implements OLS regression



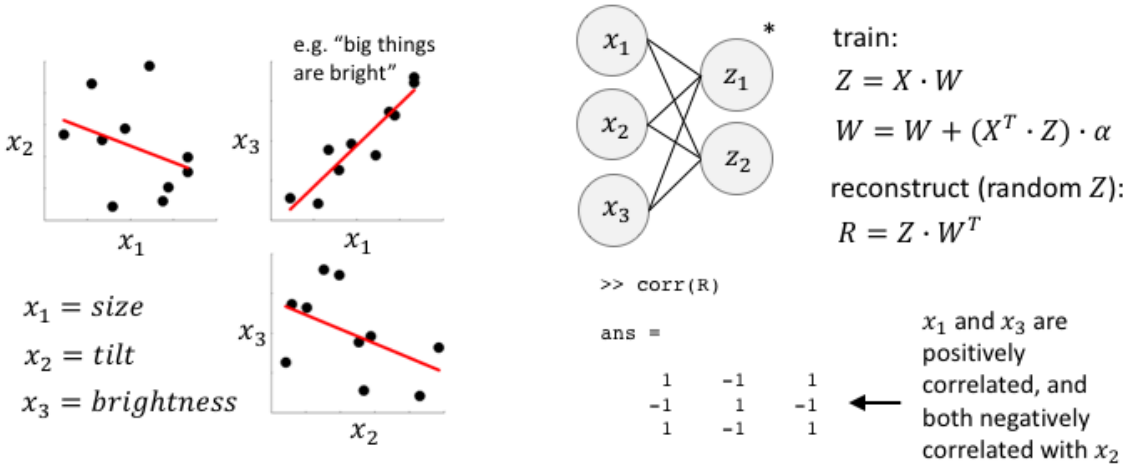
The goal is to learn the principal sources of variation in the data, providing a compact representation

e.g. map image pixels → objects

Hebbian learning directly implements PCA (Oja's rule)

Recall also that when considering feedforward neural networks we mentioned that supervised learning implements an online form of multivariate regression, in which the goal is to learn the network parameters that map an input x onto an output y through gradient descent. By analogy, simple unsupervised networks implement a form of principal components analysis (PCA), which seeks to find a lower-dimensional representation of the input data that preserves the major sources of variation in compact form. In fact, it can be shown that Hebbian learning implements an online version of PCA, also known as Oja's rule[84].

[84] Oja, E (1982). Simplified neuron model as a principal component analyzer. Journal of Mathematical Biology. 15 (3): 267–273.



The network has orthogonalised the inputs, i.e. it understands that x_1 and x_3 are positively related, and negatively related to x_2

*actually for this example only a single z unit is needed

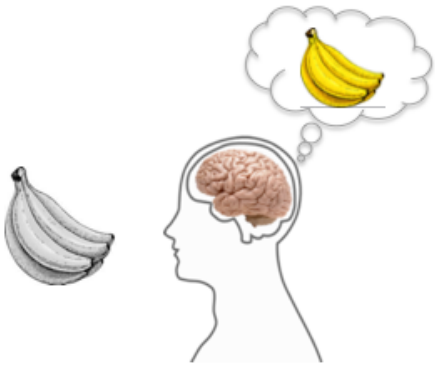
We previously saw that an autoassociative network learned by updating the weights fractionally by the outer product of the input data with its transpose $X \cdot X^T$. To implement Oja's rule, we posit a new layer of output neurons Z which is equal to $X \cdot W$, and update the weights with $X^T \cdot Z$ scaled by a learning rate. In the example on the slide, I have generated 3 noisy variables x that are partially correlated, that might for example pertain to the size, tilt and brightness of a set of objects. After learning, the patterns of correlation are encoded in the weights linking sensory inputs X to the latents Z, and for any random activation of Z we can reconstruct these correlation patterns through multiplication with the transposed weights. The network has learned, for example, that "big things are bright".

How do we know if our unsupervised network is any good? It isn't predicting some ground truth, or harvesting reward, so we don't have an evaluation metric!

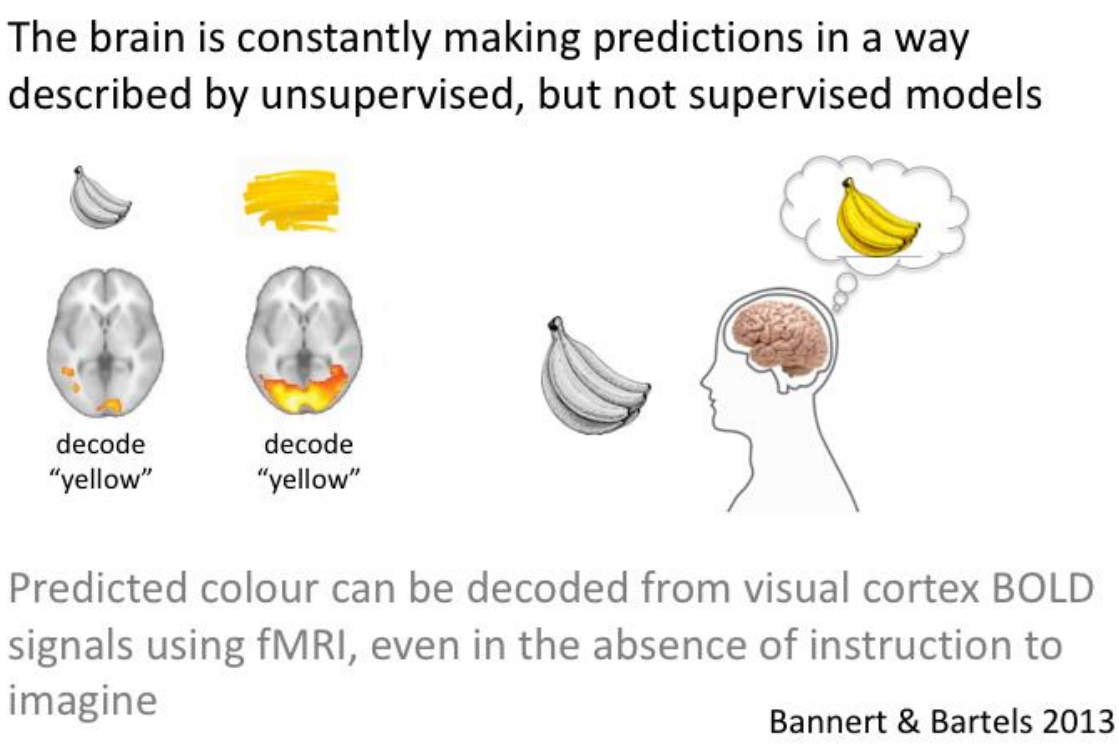
One way of thinking about an unsupervised network is as a **predictive model**

recall that $corr(x_1, x_3) = 1$

so for any new x, if I know x_1 I can predict x_3
For example, "if it is big, then it is bright"

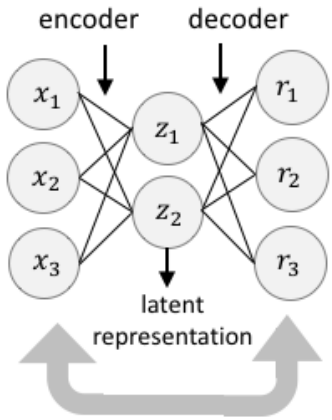


As mentioned above, an unsupervised network does not explicitly learn to map an input onto an output. Rather, in re-encodes its inputs in a new, compressed form. As such, there is no immediately obvious metric for whether a network is encoding information a "correct" or an "incorrect" way that is comparable to the decisions made by a supervised network. Nevertheless, it is possible to test the efficacy of the encoding by using the unsupervised model as a predictive network, by holding out (after training) a subset of the inputs and asking if the network can reconstruct them correctly. For example, on presented with a greyscale image of a bunch of bananas, the network should be able to correctly "predict" that they are yellow, by "decoding" information from the latent units.

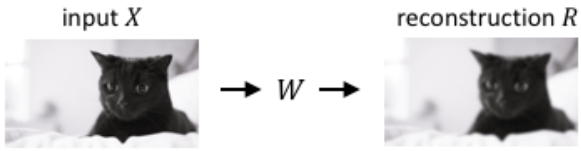


Interestingly, this is – of course – exactly what biological brains are doing. The human brain is forever making predictions about forthcoming sensory events, even when there is no particular reward that is likely to accrue from doing so[85]. Here is some evidence from an imaging study that dovetails with the example given above: participants were shown a series of greyscale images of objects with a clear associated colour (such as a banana). Using a separate localiser, the authors identified patterns of activity that were elicited when viewing different colours (such as yellow). They then asked whether the associated colour patterns were reinstated by the relevant greyscale images – for example whether the grey bananas elicited a pattern for "yellow". They found that they did, as if participants were implicitly "predicting" the missing colour of the objects.

[85] This point has been made in so many ways that it's hard to know what to cite, but two books that particularly influenced me some years ago were I of the Vortex: From Neurons to Self by Rodolfo Llinas, and On Intelligence by Jeff Hawkins, who later went on to found *Numenta*, a notable AI startup company.



$$loss = \sum_{x,y}\left[I(x,y) - \sum_{i} a_i \phi_i(x,y)\right]^2$$



Optimise W to minimize reconstruction error

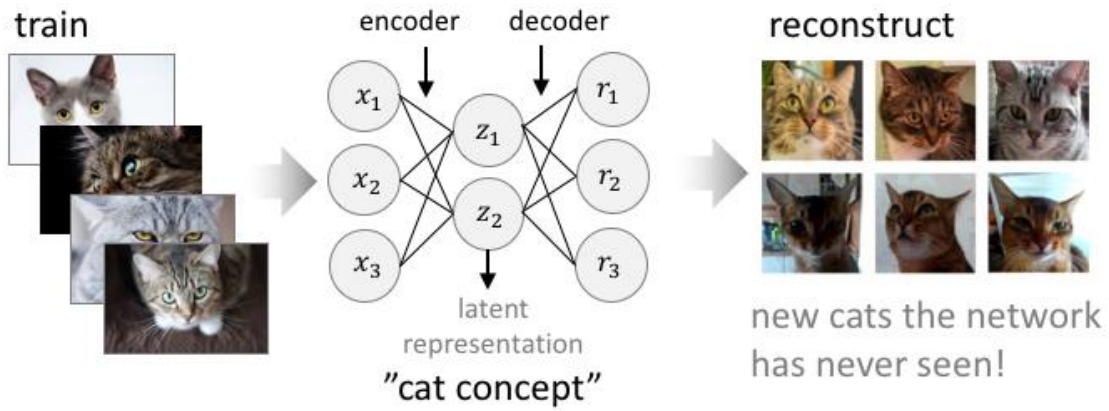
But hold on... $X \cdot I = X$!

$$\begin{bmatrix}2\\1\\3\\7\end{bmatrix} \cdot \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} = \begin{bmatrix}2\\1\\3\\7\end{bmatrix}$$

So why doesn't the network just learn the identity matrix?

For the network to compress the image, $n(Z) < n(X)$ i.e. the latent layer must be a "bottleneck"

Many of the most successful contemporary unsupervised learning methods are based around the notion of "autoencoding". A network is trained on a set of images, and the loss is the reconstruction error, i.e. a quantity that is inversely proportional to the network's ability to predict the very image it is being shown. Of course, there is a trivial way to solve this problem – if the network weights converge to the identity matrix (zeros everywhere and ones of the diagonal) then every output will be identical to its input. But remember that the point of unsupervised learning is to reduce the dimensionality of the input whilst preserving as much of the variance as possible. Thus, the number of hidden (or "latent") units, typically denoted Z must be smaller than the number of input units, precluding the network from learning identity weights. This forces the network to learn an efficient code.



The goal is to build an generative network that can "imagine" realistic images, text, music… just like humans can write new poems and compose new novels…

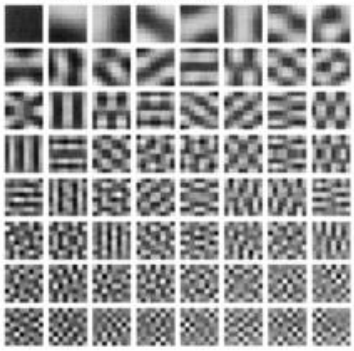
In fact, if the network successfully captures the true data-generating latent variables, then it should be possible to decode from the network new images that correspond to realistic data samples. In other words, after training the network to encode images of cats, it should be possible by randomly activating the latent units to decode realistic-looking images of new cats, as if the network were "imagining" what cats might be possible. In other words, the network has learned a "generative model" of cats and can generate cats. The human cognitive ability to imagine and mentally simulate the environment, thus, relies on learning such a "generative model" of the world.

OK. so why not just do PCA on image pixels?

You could. But you don't learn a set of biologically plausible filters (basis functions), and the reconstructions don't resemble real images

PCA captures the structure of data in which linear pairwise correlations are the most important form of statistical dependence

filters from PCA model



$$I(x,y) = \sum_i a_i \phi_i(x,y)$$

x and y here denote image axes, not network inputs/outputs

Olhausen & Field 1996

So how do you actually do this? Well, one approach would simply be to use Hebbian learning, in other words to try and learn the principle components of natural images. In practice, however, this doesn't tend to yield very good predictions about new images, and the network learns a set of filters that are quite different from those observed in biological visual systems.

However, a sparsity assumption helps

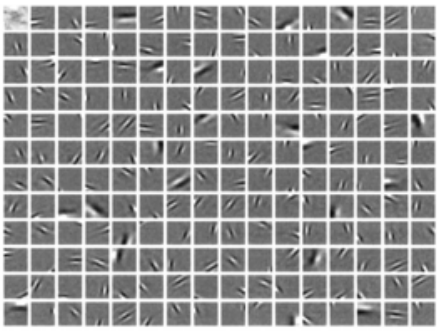
$$loss = \sum_{x,y} \left[I(x,y) - \sum_i a_i \phi_i(x,y) \right]^2 + log\left[1 + \left(\frac{a_i}{\sigma}\right)^2\right]$$

reconstruction error

sparsity cost

The sparsity cost ensures that optimization finds the solution with the smallest number of nonzero coefficients (weights)

filters from sparse coding model



This allows biologically plausible basis functions to be learned

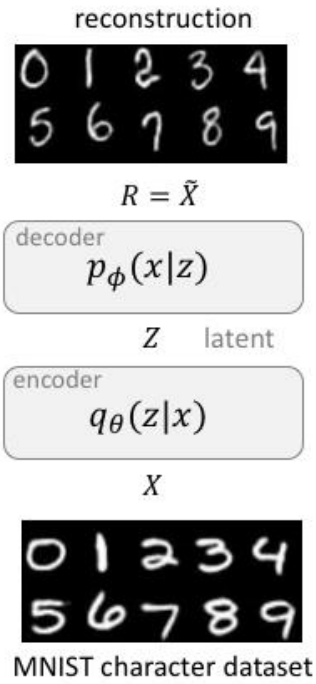
i.e. orientation and spatial frequency selective receptive fields such as those observed in V1

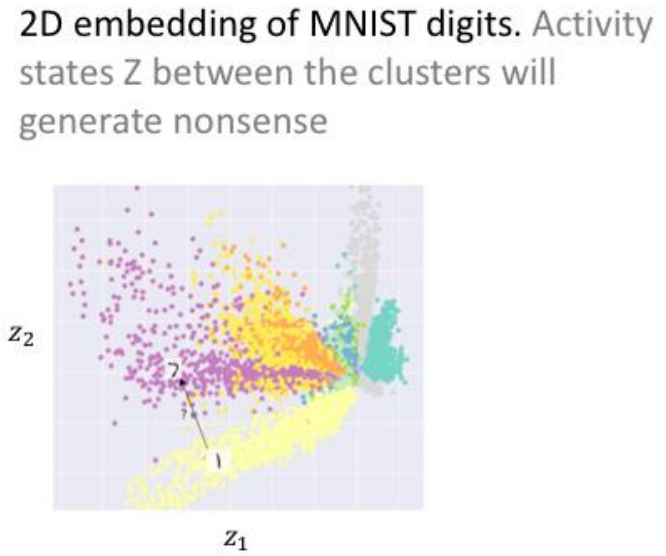
However, reconstruction is still poor

Olhausen & Field 1996

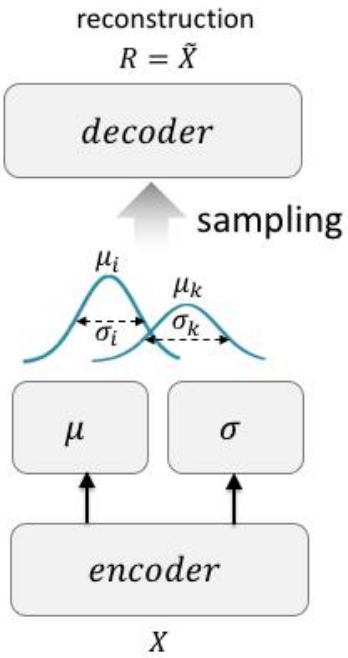
One way to improve the filter quality is to impose a sparsity constraint, i.e. to add a term to the loss function that encourages as few neurons as possible to have nonzero weights. Using this additional sparsity cost, an unsupervised network trained on natural images will learn components that correspond to a set of Gabor filters with varying orientation and spatial frequency, just like cells in V1. However, although reconstruction of trained images is reasonably good, this class of network shows limited ability to "imagine" realistic new images from the full distribution of natural scenes.

7.3. Variational autoencoders

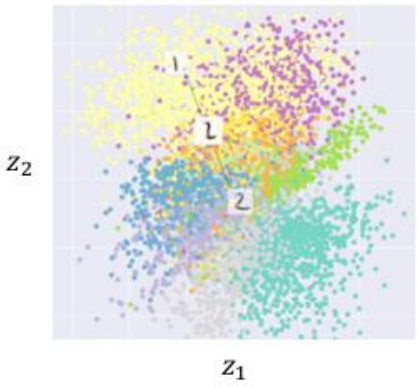




However, with the addition of some further computational tools, it's possible to build a network that can learn the distribution over reasonably complex naturalistic inputs, such as handwritten digits. The variational autoencoder uses several stacked encoding/decoding layers.



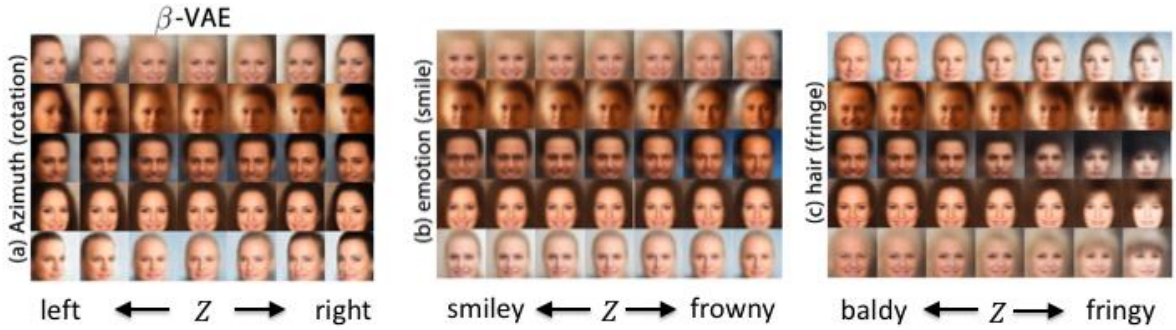
Now there's a much better latent representation. Note that 7 (purple) lies nearly between 1 and 2



The VAE uses an additional cost term that tries to keep the parameters $\mu \sim 0$ and $\sigma \sim 1$ i.e. keep the sampling distributions standard normal

However, the key differences is that what is encoded in the hidden units are values corresponding to the mean and variance of the latent variables. The decoder then samples from these distributions to generate new data. The VAE[86] incorporates a distinct cost (KL divergence) that attempts to keep the sampling distributions as close as possible to standard normal (e.g. mean = 0, std = 1). That means that the data distributions are smoothly distributed over the latent space, permitting new data to be generated from this space through interpolation (without this KL penalty, the there are gaps in the latent space, and sampling from these areas will produce nonsense images).

[86] This paper has nice explanations: https://arxiv.org/pdf/1606.05579.pdf. Also see this website: http://kvfrans.com/variational-autoencoders-explained/ and this tutorial: https://arxiv.org/abs/1606.05908.



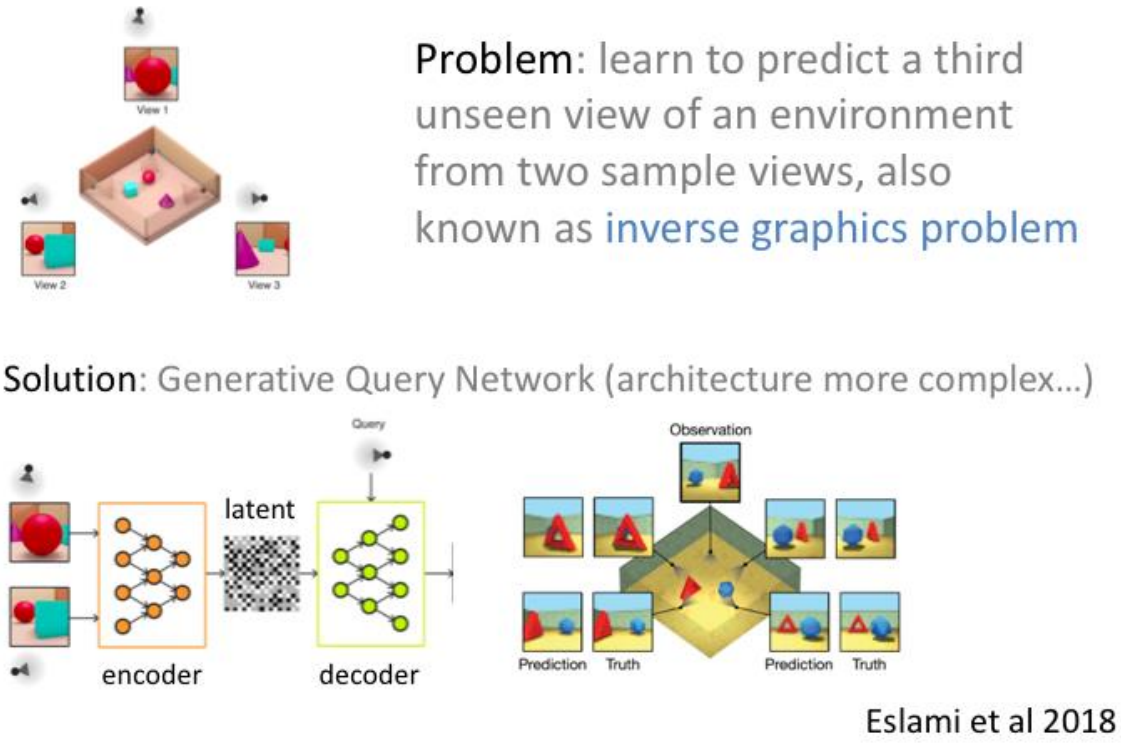
The β-VAE simply places a stronger constraint on the latent bottleneck relative to reconstruction error

One could argue that this network has learned visual "concepts" pertaining to faces and other objects

Higgins et al 2017

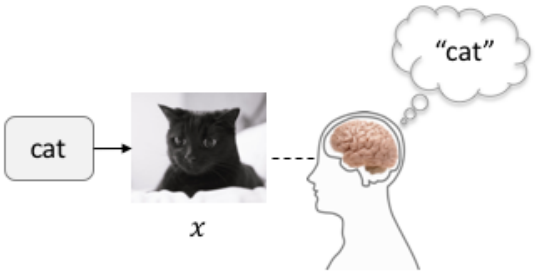
In fact, it was recently discovered that by increasing the strength of the KL penalty, in order to place greater pressure on the network to learn sensible latent variables, it is possible to learn human-interpretable factors of variation even from complex high-dimensional domains such as faces. The slide shows reconstructions from a so-called β-VAE[87] which incorporates this penalty (the β is the parameter that controls the relative balance of reconstruction error to the KL divergence). By gradually varying activation in the latent units and reconstructing, it is possible to generate new faces with varying degrees of rotation, smile and hair coverage, as if the network has learned some of the "true" factors that determine how faces vary.

[87] https://openreview.net/pdf?id=Sy2fzU9gl



A yet more powerful generative model, based on similar principles, has recently been applied to a longstanding problem in computer vision, know as the inverse graphics problem: given a small number of snapshots taken from a 3D environment, can you "imagine" the perspective from a new angle? The network was trained on observation data and camera angle taken from complex synthetic images showing tabletop objects and mazes, and trained to predict the new viewpoint conditional on the camera angle. Using a deep generative model with a similar form to the autoencoder discussed above, the network was able to do this with a high degree of accuracy.

7.4. The Bayesian approach



Let's simplify and propose the existence of neurons that directly code the inputs $\hat{x}$

Encoding model: compute $p(\hat{x}|x)$, i.e. the probability that a neuron is active, given the input, and combine with prior $p(x)$

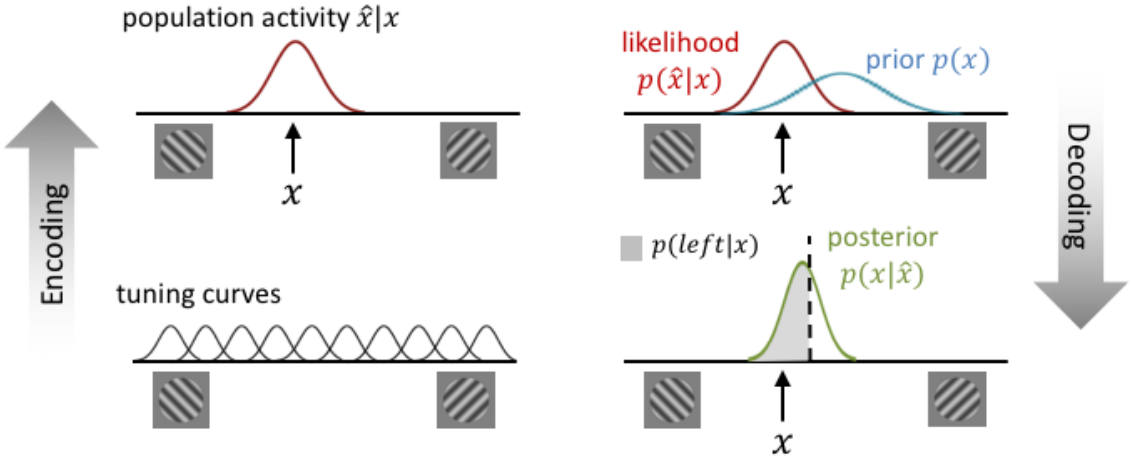
Decoding model: compute $p(x|\hat{x})$ and marginalize by decision criterion to obtain $p(cat|x)$

According to Bayes' rule, the posterior is proportional to prior x likelihood

$$p(cat|x) \propto p(cat) \times p(x|cat)$$

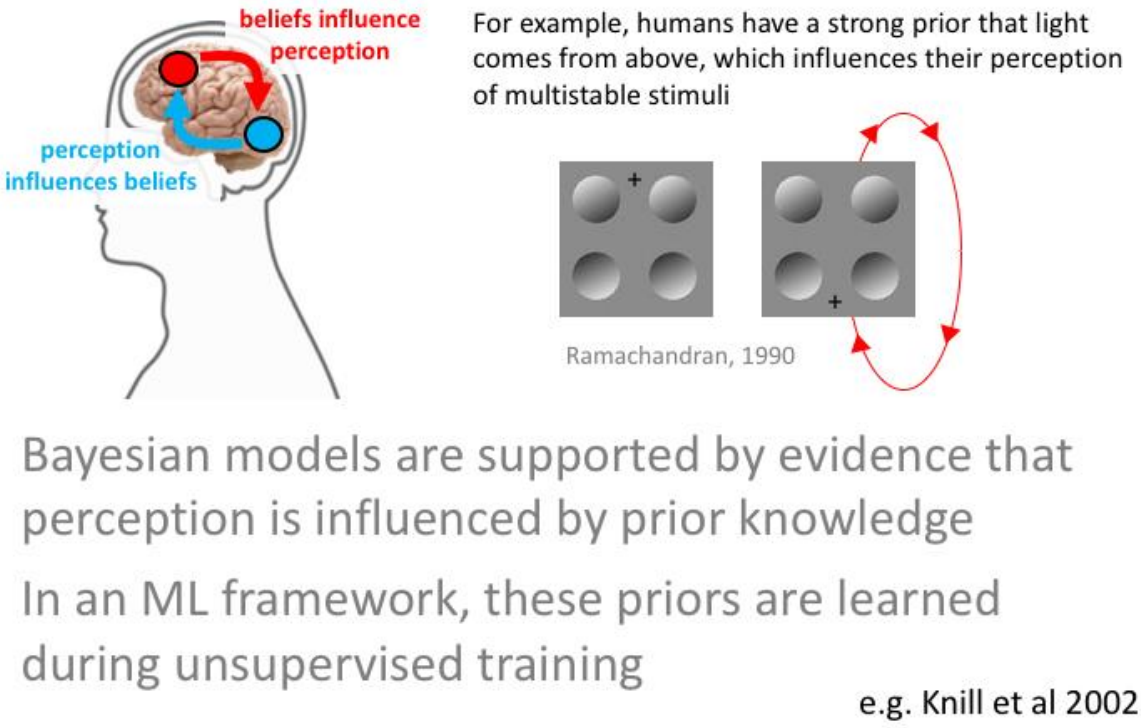
A long tradition in psychology considers perception as a inference problem. Optimal solutions are given by Bayes' rule

The unsupervised learning framework described here shares a theoretical stance with a long tradition in psychology that sees perception as an inference problem, with optimal solutions given by Bayes' rule. In other words, the brain evolved to encode $p(\hat{x}|x)$, i.e. the probability of a given pattern of neuronal activation conditional on the inputs, and to decode $p(x|\hat{x})$, i.e. the probability of the data given the hypothesis.

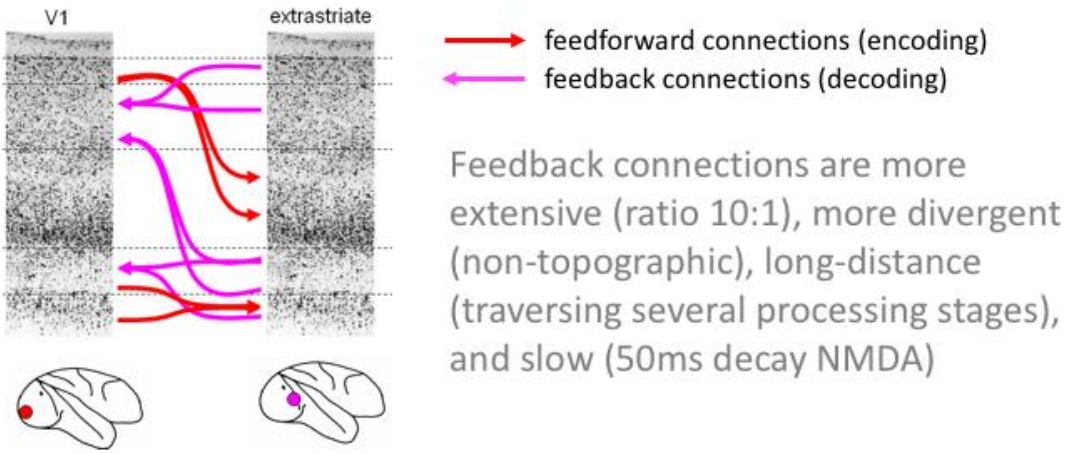


In psychophysics, we typically assume a **fixed** encoding model with Gaussian tuning curves, a simplification that works well for elementary features such as tilt

In psychology and sensory neuroscience, this approach is most often applied to simple experimental domains, such as psychophysical studies, where the state space is small (e.g. consists of a space of possible angles of orientation) and the filters are of known form (e.g. Gaussian tuning curves). In this setting, we can see the encoding step as estimating the likelihood of neuronal response given the input, i.e. providing the population activity; and the decoding step as computing some posterior distribution over input states, which depends both on the likelihood term and any relevant prior beliefs, which may accrue from top-down signals. Marginalisation on the posterior gives a probability of one response vs. another (for example, in a binary choice task).



Psychologically and neutrally, this work draws on evidence that human perception is strongly influenced by prior beliefs, such as the fact that light comes from above when interpreting shape from shading. In many models, perception involves a reciprocal interaction between bottom-up processes (whereby perceptual inputs influence beliefs) and top-down processes (whereby beliefs help shape perception).



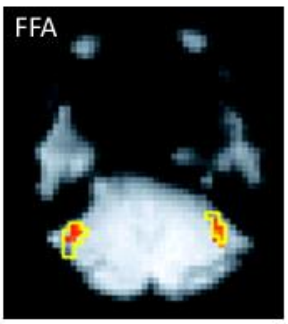
Moreover, the notion that there is an economy between bottom-up (encoding) and top-down (decoding) helps us understand why sensory systems – and in particular vision – prominently include not just feedforward but also feedback connections. In fact, in the primate visual systems, feedback connections are more prevalent than feedforward connections, which might be interpreted as supporting the "reconstructive" nature of perception.

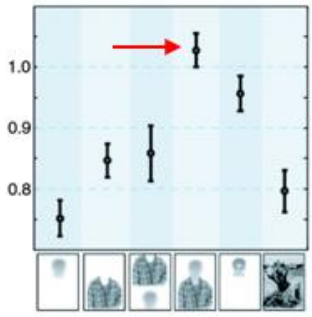
What is this?

Cloze figures show the influence that memory has on perception

e.g. Knill et al 2002

Many other findings support the view that perception is a reconstruction that incorporates our prior beliefs. For example, on viewing the upper image for the first time, you might struggle to interpret it; but having seen the image below, it becomes immediately apparent what it is.



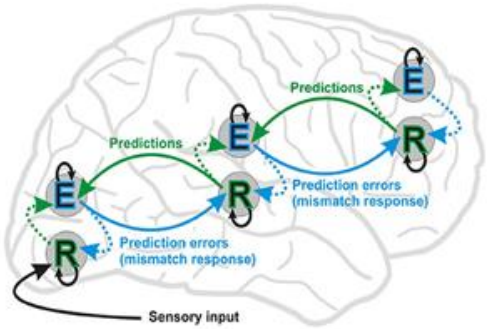


FFA BOLD responds more strong to the suggestion of a face than to the actual presentation of a face

Cox et al 2004

Further evidence, akin to the banana example above, comes from imaging studies. For example, in this striking fMRI study, the FFA was found to be more active to a nonface that resembled a face by virtue of the context than it was to a decontextualized actual face.

7.5. Predictive coding



prediction ~ decoding

prediction errors ~ encoding (of residuals)

Predictive coding is a computational framework proposing that encoding and decoding occur repeatedly at successive hierarchical levels of the cortex

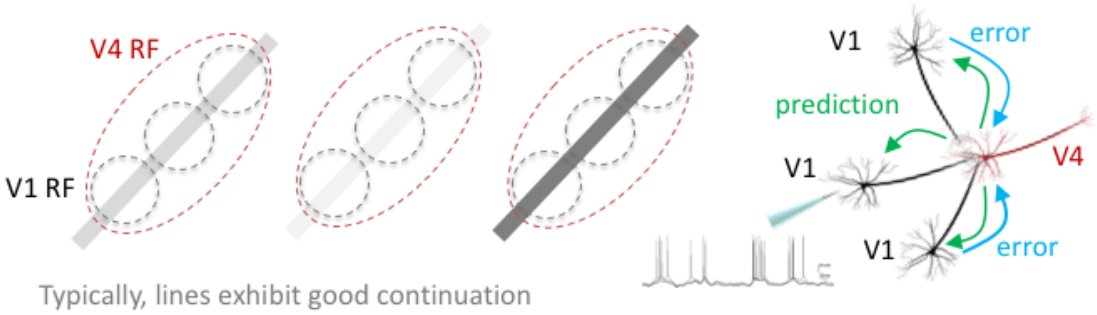
Priors at each level are combined with sensory signals at the level below, via re-entrant (feedback) connectivity

Error signal are computed as the mismatch between predictions and error signals, and are sent forward to adjust future predictions

Predictive coding proposes that perceptual inference occurs via multiple iterative cycles of encoding and decoding

Bastos 2012; Friston 2005

A mature computational framework, known as predictive coding, argues that perception unfolds in successive layers of the visual hierarchy, with prior beliefs feeding back via top-down signals to "explain away" sensory inputs at each stage, such that only the unexplained portion of sensory signals (sensory prediction errors) are passed forward to adjust beliefs. This theory explains a variety of interesting phenomena, including the fact that BOLD signals and single-cell responses tend to be particularly strong when inputs are unexpected. In fact, one paper argues that repetition suppression, the ubiquitously observed attenuation of neural signals to the second and subsequent presentations of a stimulus, may be in fact be a reduction in sensory prediction errors in under the predictive coding scheme.



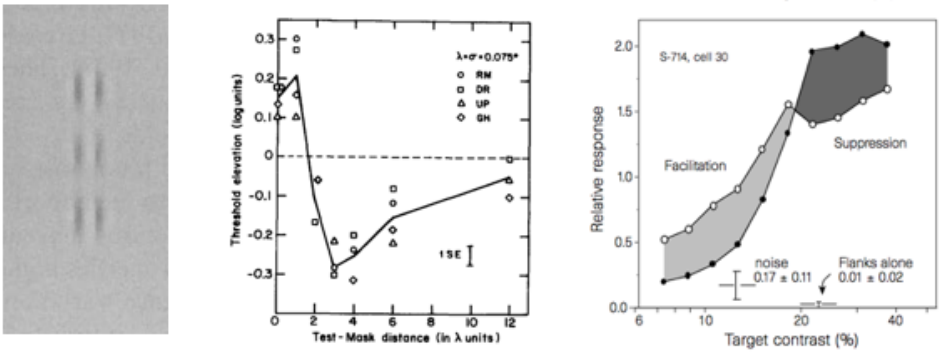
Typically, lines exhibit good continuation

When the line is low-contrast, the response in V1 is facilitated by colinear flankers, due to predictive signals

When the line is high-contrast, the V1 response is suppressed, as inputs are "explained away" by predictive signals.

Rao & Ballard 1999

Predictive coding offers an elegant account of extra-classical receptive field effects, such as end-stopping, and also of classic visual phenomena, such as contextual facilitation.

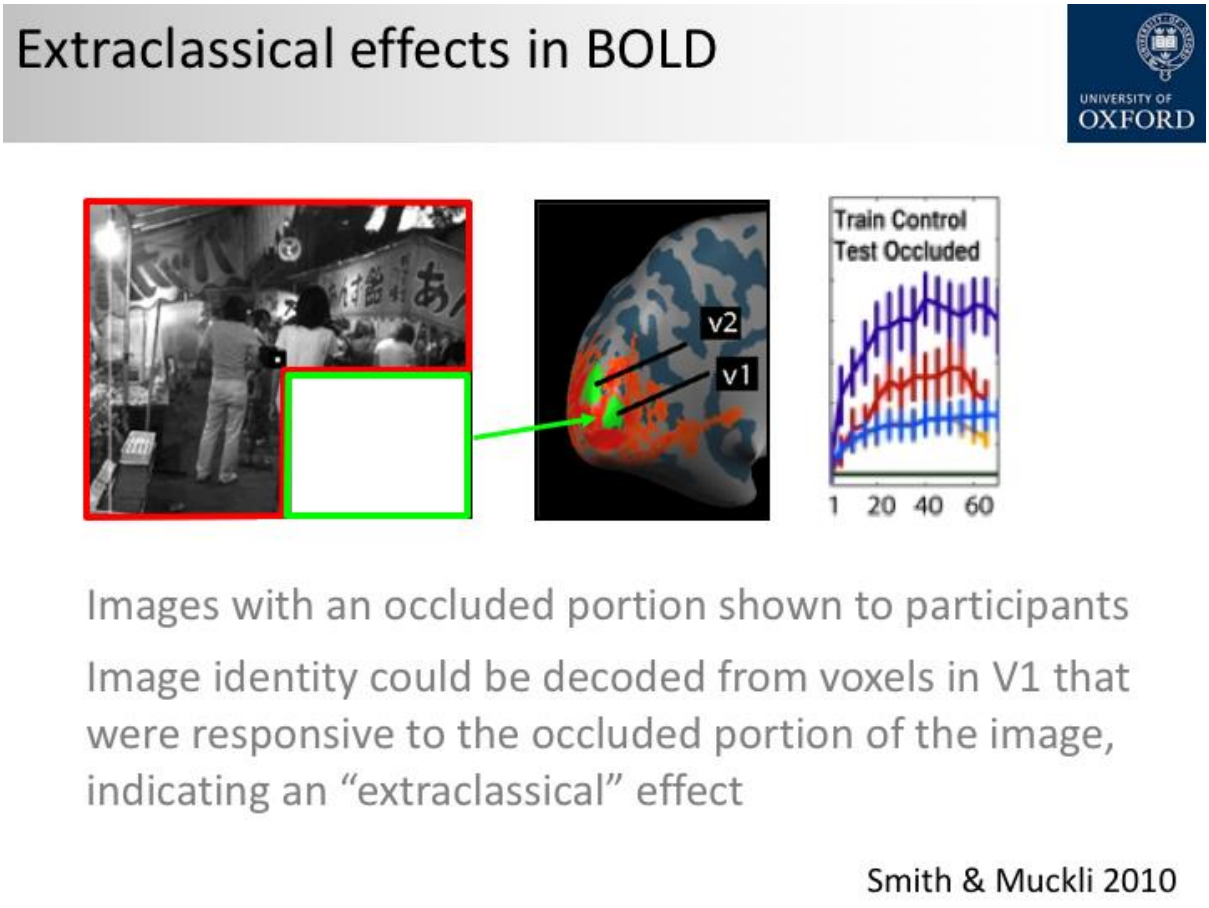


Lateral "masking" stimuli decrease discrimination threshold for a central grating (contextual facilitation)

Neural responses exhibit the facilitation-suppression pattern of responses hypothesised by predictive coding

Polat & Sagi 1993; Polat 1998

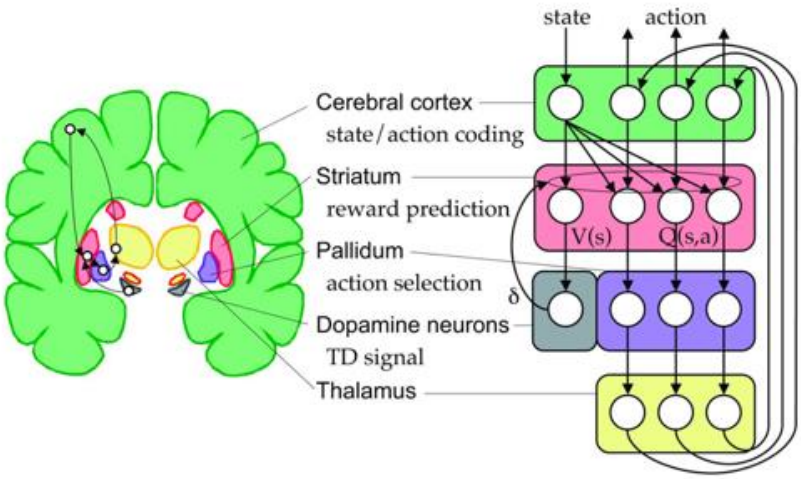
Polat and Sagi have provided data supporting this pattern of contextual facilitation, as hypothesised by predictive coding, from elegant psychophysical work.



Finally, we can even see evidence for extraclassical effects in BOLD signals. In this paper from Smith & Muckli, participants viewed a series of natural scenes with and without an occluded quadrant. The authors first identified voxels that were retinotopically mapped to the occluded portion of the image, and then asked whether, from these voxels alone, it might possible to decode the contents of the image. They found that it was. The only reasonable explanation for this finding is that those voxels are receiving predictive feedback signals from other brain areas, for example more anteriorly, that have access to the image information via neurons mapped to other portions of retinotopic space.

8. Building a model of the world for planning and reasoning

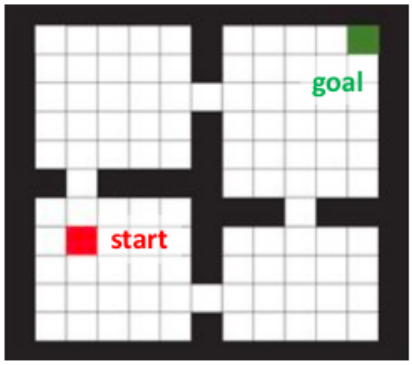
8.1. Temporal abstraction for model-free RL and the dACC



General models proposed for mapping computations of RL onto circuitry of basal ganglia

Doya, 2007

Let's begin by taking a step back to lecture 2, where we discussed model-free RL. You'll recall that we converged on a circuit-level description of how model-free RL might be implemented in biological brains, via parallel circuits linking cortex and striatum, with dopaminergic signals carrying a TD error signal that allows connections for rewarded actions to be strengthened.



Consider the following "four rooms" problem. How well is model-free RL going to do?

A. not very well, because it will take a long time to reach the goal initially

Model-free RL scales poorly to large environments with sparse rewards

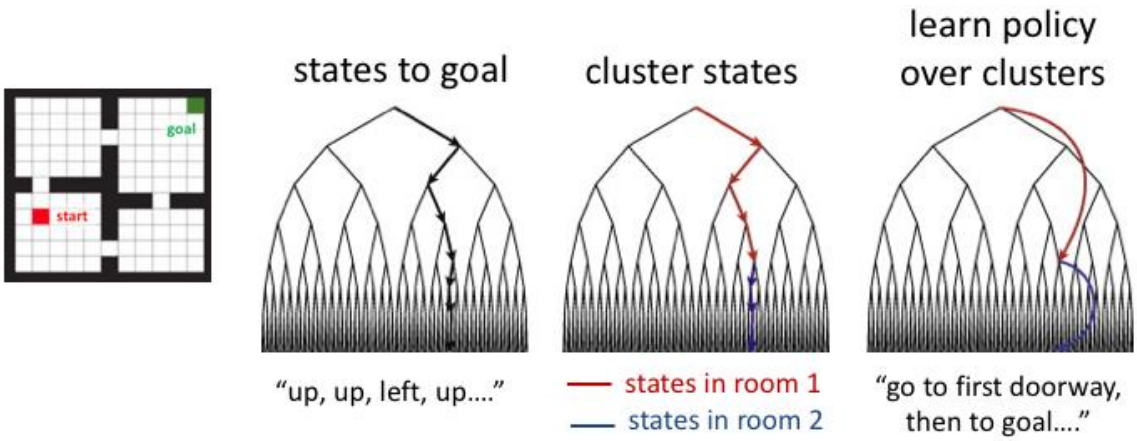
One solution is temporal abstraction

Botvinick et al 2009

However, these "model-free" reinforcement learning methods have some important limitations. Consider the environment shown on the slide above. The agent starts from the red location and has to reach the green. How well is it going to do? Well, we know that (for example) TD learning approximates the Bellman equation, so it should eventually learn the optimal value function. But this is going to be very, very slow, because to get to the goal it has to pass through 2 doorways under an essentially random policy (no knowledge of the value of actions). In general, model-free RL scales very poorly to environments (like the real world) where there are innumerable states, and rewards are sparsely distributed. In the final part of this lecture, we will consider some methods that have been developed to deal with this problem, that are focussed around the idea of *temporal abstraction*[88].

2.7. Hierarchical reinforcement learning and temporal abstraction

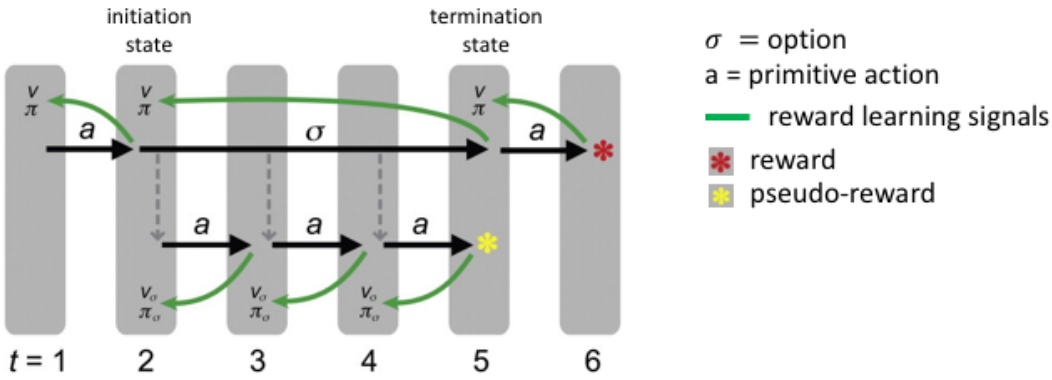
[88] Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. Botvinick MM, Niv Y, Barto AC. Cognition. 2009 Dec;113(3):262-80.



Temporal abstraction is possible when states are can be meaningfully clustered in time. For example, many environments can be encoded hierarchically.

Botvinick et al 2009

Temporal abstraction is made possible when states can be meaningfully clustered in time. Consider, for example, the "four rooms" environment illustrated above. Whilst there are many ways that the trajectory to goal might be represented, not all of them are equally useful or compact. One particularly efficient code represents the trajectory hierarchically, much as a human might instruct another: go to the doorway, and from thence to the second doorway, and then to the goal. If an agent could select not just among primitive actions (up, down left, right) but also among chunked action sequences ('go to the doorway'), then learning would proceed much faster.

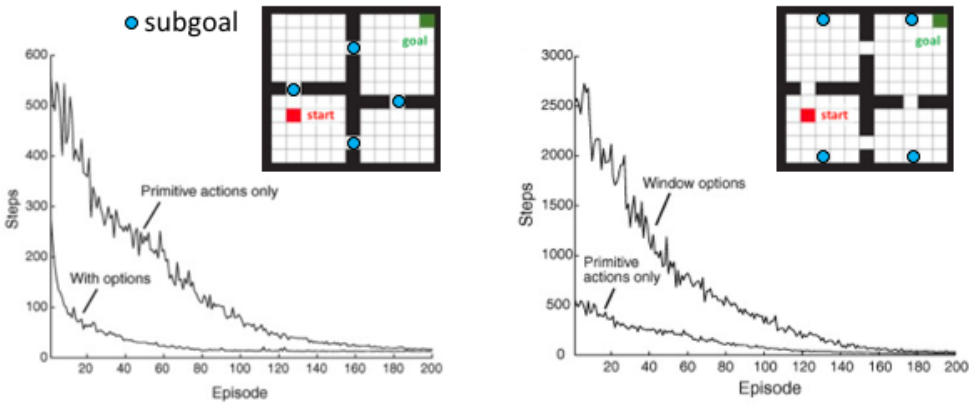


In HRL, a special set of states are pre-defined where an **option** may be initiated or terminated (e.g. a doorway)

In initiation states, and option may be selected instead of a primitive action. The option specifies a series of actions that are executed until termination e.g. {left, left, up, left}

Botvinick et al 2009

This is the principle by which "hierarchical reinforcement learning" (HRL) works. Let us assume, for the sake of argument, that a set of key states in the environment are pre-designated as having special status, by virtue of their importance for any given plan (such as the doorways in the 4-rooms environment). We will call these states "subgoals". These subgoal states are earmarked as those where an *option* may be initiated or terminated. An option specifies a series of actions that are executed until execution. Options are learned based on an intrinsic reward signal – known as a pseudoreward – that is emitted when a subgoal is reached. The agent can thus first learn to reach a subgoal (and receive a pseudoreward), and can then at future timepoints opt to select the entire set of actions that will take it to the relevant subgoal. The figure above illustrates the computations executed by an HRL agent as it makes a set of 6 transitions through an MDP. On step 2, it reaches an initiation state and selects option σ, making primitive actions a until the termination state is reached on step 6 and a pseudoreward (yellow star) is emitted. This pseudoreward is backed up to increase the value of the primitive actions. On the next step, the agent reaches the goal, and a real reward is incurred; this is backed up to the option selection stage, increasing the probability that this option will be selected in the future.

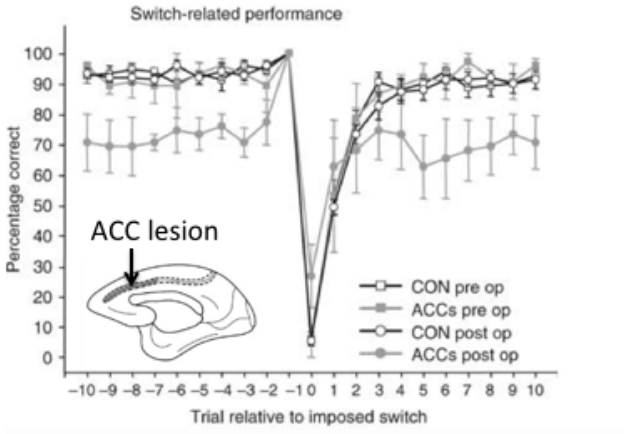


HRL dramatically accelerates learning in on the four-rooms problem, relative to standard actor-critic methods, as long as the subgoals are in the right place!!

The subgoal discovery problem is unsolved…

Botvinick et al 2009

HRL works, subject to some caveats. For example, an RL agent learning to navigate to a goal location in the four rooms environment learns faster with options than without. However, one critical aspect of HRL is that the subgoals need to be appropriately prespecified. For example, if the subgoals are placed not at the doorways but at the "window" locations shown in the right-hand plot, the HRL agent does worse than an agent without options. More generally, the problem of how the agent (magically) knows which states should be subgoals is unsolved, dramatically limiting the utility of HRL in real world settings.



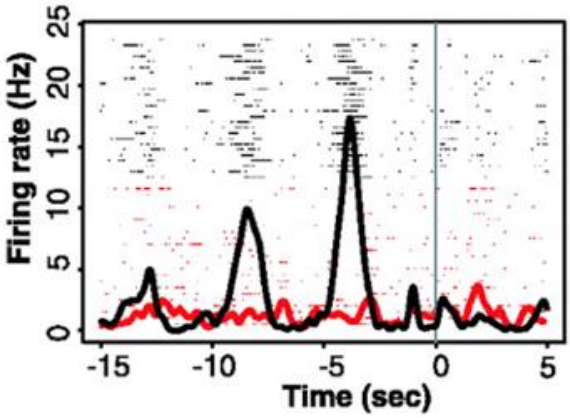
dACC lesions provoke failures of sustained action selection in a reversal learning task, i.e. in ability to stick to a fixed policy

Neural evidence supports a role for the dACC in HRL

See Holroyd & Yeung 2012

Kennerley et al 2006

Nevertheless, there is neural evidence that animals engage in a form of sequential actions selection not dissimilar to that proposed by the options framework, and that its implementation depends on the integrity of the dorsal anterior cingulate cortex (dACC). One key assumption of the options framework is that a specific mechanism exists that ensures that the option is followed once initiated (at least when it is reasonable to do so) rather than other primitive actions that are not specified by the option being selected. Lesions of the dACC make macaque monkeys more likely to switch away from a consistent course of action, such as selecting a response with high reward probability in a reversal learning task, as if they were less prone to pursue a fixed course of action (the same lesions have little or no effect on the animals' tendency to reverse at the correct time, relative to control animals)[89]. Holroyd and Yeung[90] have suggested that implementing sequential action control in a manner similar to that proposed by HRL model may be the cardinal function of the dACC, with similar views proposed by Botvinick[91].



dACC neurons signal proximity to a reward during extended behaviours (black line)

This suggests that dACC neurons signal proximity to the termination of an option

Shidara & Richmond 2002

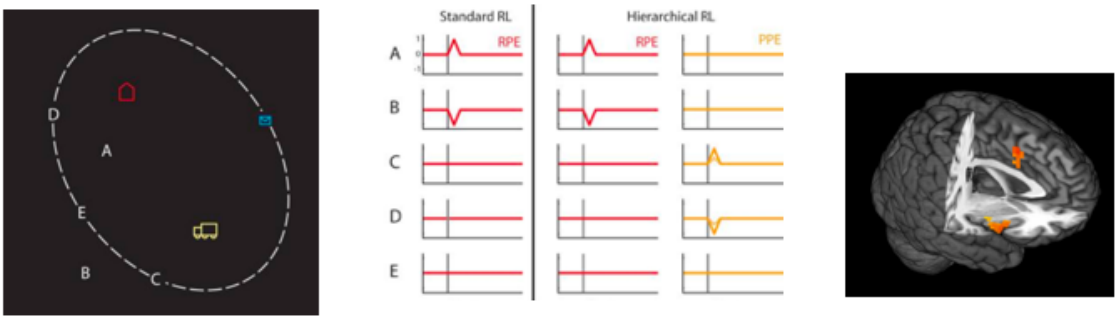
One further prediction that arises from the options framework is that the agent needs to monitor for the presence of a termination state, at which point the option is no longer followed and a new action can be selected. Shidara and Richmond (2002) showed that when multiple sequential actions need to be made to elicit a reward, dACC neurons code for proximity to the

[89] Optimal decision making and the anterior cingulate cortex.Kennerley SW, Walton ME, Behrens TE, Buckley MJ, Rushworth MF. Nat Neurosci. 2006 Jul;9(7):940-7

[90] Motivation of extended behaviors by anterior cingulate cortex. Holroyd CB, Yeung N. Trends Cogn Sci. 2012 F

[91] Hierarchical reinforcement learning and decision making. Botvinick MM. Curr Opin Neurobiol. 2012 Dec;22(6):956-62.

sequence termination[92]. Others have shown that the dACC BOLD signals are sensitive to the proximity to a switch point in a foraging setting.



Delivery problem: drive to package, then to house. But package can jump to locations A-E

HRL predicts pseudo prediction error (PPE) for jumps C and D, but standard RPE for A and B

dACC BOLD correlates with PPE

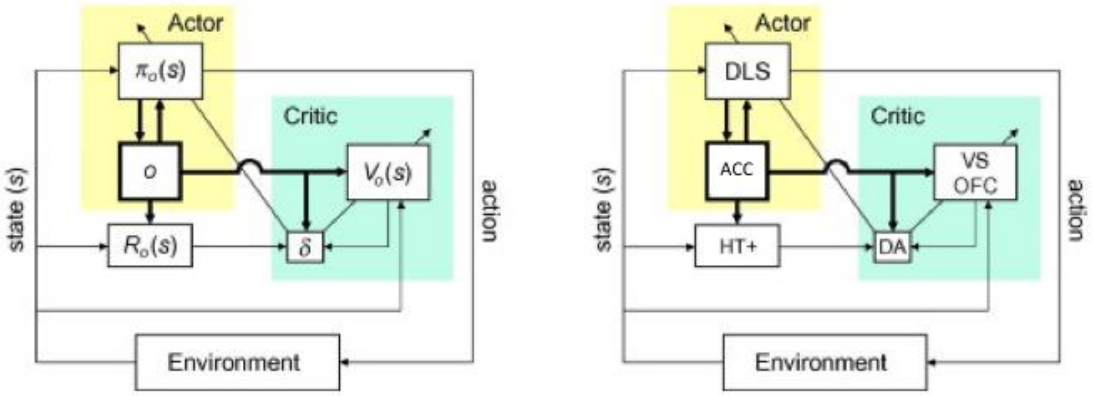
This fMRI study shows dACC correlates with "pseudo prediction errors" obtained when a subgoal changes value

Ribas-Fernandes et al 2012

Further evidence for the role of the dACC in an HRL-like process comes from an imaging study[93] that directly tested for the pseudoreward signals that are predicted by HRL. The authors designed a task that involved navigating first to a subgoal location, where no reward was incurred (driving a truck to pick up a parcel) and then to a goal location for reward (on delivery of the parcel to a second location). The authors induced pseudo-prediction errors by switching the subgoal in such a way that it neither shortened nor lengthened the overall trajectory but increased or decreased the subgoal distance. These pseudo prediction errors were correlated with BOLD signals in the dACC.

[92] Anterior cingulate: single neuronal signals related to degree of reward expectancy. Shidara M, Richmond BJ. Science. 2002 May 31;296(5573):1709-11.

[93] A neural signature of hierarchical reinforcement learning. Ribas-Fernandes JJ, Solway A, Diuk C, McGuire JT, Barto AG, Niv Y, Botvinick MM. Neuron. 2011 Jul 28;71(2):370-9.

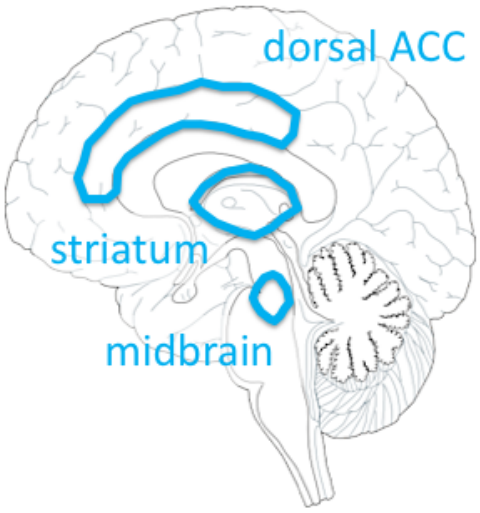


DLS = dorsolateral striatum VS = ventral striatum OFC = orbitofrontal cortex
HT+ = hypothalamus and related structures ACC = anterior cingulate cortex

Together, these findings suggest a circuit mechanism for implements RL in the mammalian brain

Botvinick 2008

Bringing all these finding together, thus, we can expand our model of how RL is implemented in the brain, to include new modules that engage in sequential control over actions. These may be associated with the dACC, although other regions such as the DLPFC may also play a part.

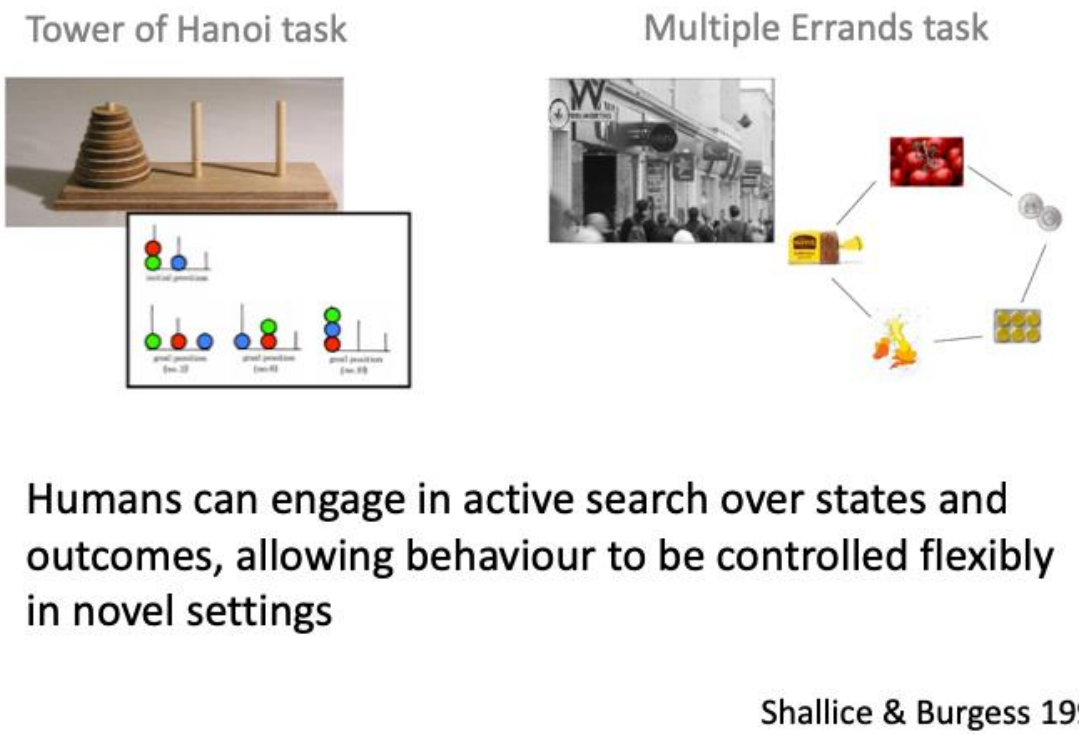


The striatum may learn action values through via DA gating of sensorimotor Hebbian learning, according to an RL process

The dACC participates in temporally extended action selection, and may implement a hierarchical learning process

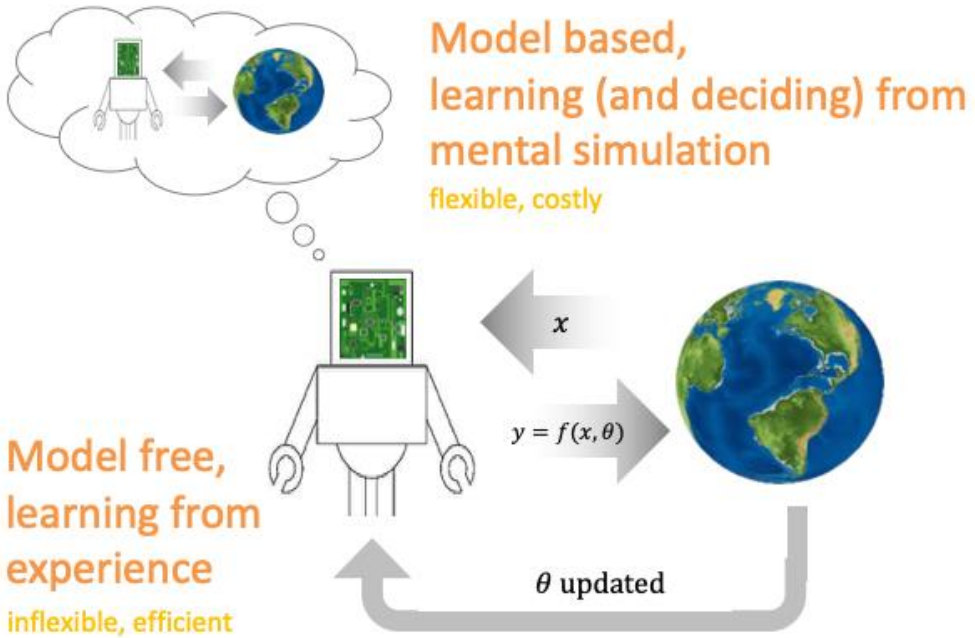
So, we began with these structures as candidates for a brain system that implements action selection. We have seen that the striatum codes the value of actions, as predicted by TD learning, via a gating signal from midbrain dopamine neurons. The dACC facilitates extended action selection, in a way that resembles HRL.

8.2. Multiple controllers for behaviour



However, there may be other mechanisms, beyond model-free RL, by which biological agents select temporally extended actions in complex environments. Humans, and probably other animals, do not have to repeatedly sample the world in order to formulate plans of action. Indeed, many environments do not permit the sort of slow learning by experience that occurs during model-free RL. For example, when deciding whom to marry, or what career to pursue, you don't generally get to try out hundreds of options and learn gradually what works and what doesn't. Rather, you have to select a course of action by engaging in mental simulation (or planning) – to consider what the likely outcomes of behaviour might be by imagining their consequences. In humans, planning is particularly associated with the functioning of the intact prefrontal cortex. When the PFC is lesioned, humans tend to show disordered patterns of everyday behaviour – for example, they will fail to wash or pay their taxes, they act impulsively or in a socially inappropriate fashion. Shallice and Burgess[94] captured these behaviours by constructing tasks that were based outside the laboratory and resembled everyday activities, such as shopping. In their multiple errands task, PFC patients failed to follow a simple set of instructions on a shopping trip (such as, "buy a newspaper", "don't enter the same shop twice", and "don't steal anything"). PFC patients also perform worse than controls on lab-based tests of planning, such as the Tower of Hanoi task, in which a set of disks have to be rearranged onto three poles without a larger disk ever lying on top of a smaller one.

[94] Shallice T., Burgess P. W. Deficits in strategy application following frontal lobe damage in man. Brain. 1991;114:727–741



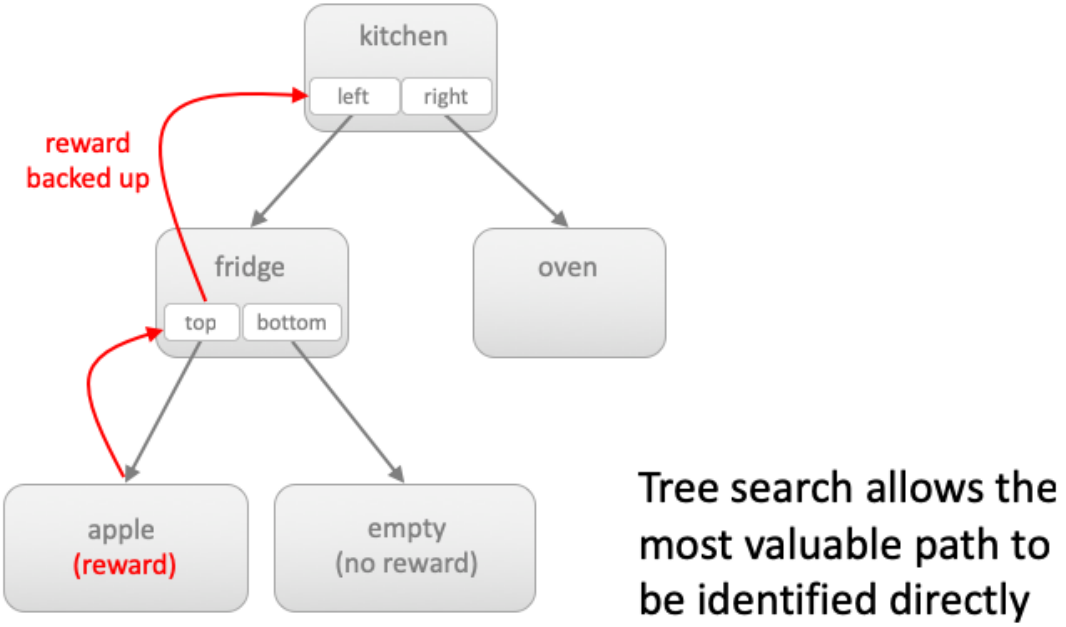
Daw & Dayan 2005; Dolan & Dayan 2013

So one way to think about behaviour in biological agents is that it is controlled by (at least) two distinct systems: one which learns slowly from experience (the model-free system) and another that allows for mental simulation (the model-based system). The model-based system is so called because planning requires a model of the world – in other words, it needs the agent to explicitly encode the transition matrix which defines how states of the world are organised with respect to one another. Full specification of a world model allows the agent to "imagine" different courses of action and their likely ultimate consequences, and to choose proximal actions that lead gradually towards rewarding outcomes, without ever having taken the chosen path before. For example, you might imagine that if you study hard then you will pass your exams and go on to receive a high-paying job and be able to buy a yacht, if you thought that owning a yacht would be a rewarding experience. Notably, you wouldn't ever have had to sat an exam, taken a job interview, or gone sailing in order to formulate this course of action – if your world model is accurate, you can use it for planning.

Critically, model-based and model free learning have complementary costs and benefits[95]. Model-free learning is fast and efficient, but it is also inflexible. Because it involves learning slowly from experience, obtaining summary estimates of the value of different states by averaging across repeated encounters, model-free estimates change only slowly. This means that if the world changes rapidly, then for a purely model-free agent, state-action value

[95] Daw ND, Niv Y, Dayan P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nat Neurosci. 2005;8:1704–1711. Ray J. Dolan, Peter Dayan. Goals and Habits in the Brain. Neuron. 2013 Oct 16; 80(2): 312–325.

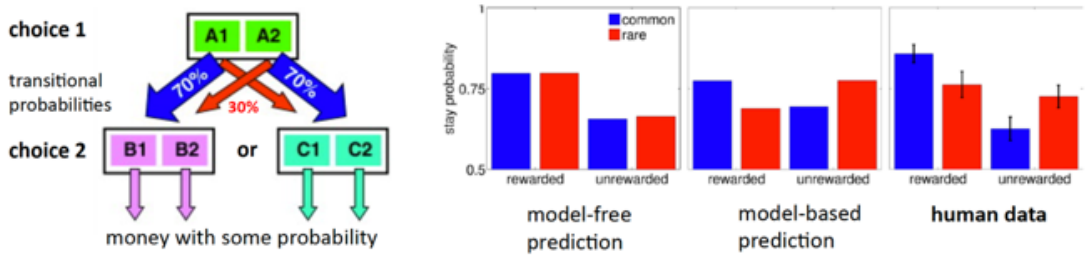
estimates may rapidly become obsolete. Model-based learning and deciding, however, is computationally costly, because it requires the agent to learn about the world (i.e. to know what state/action combinations are likely to lead to which states), and moreover, it requires the agent to search through a potentially very large space of possible outcomes. For example, it's very time-consuming to imagine what the precise consequences of every menu choice might be in a restaurant or course option might be on a university degree. It's likely, thus, that biological agents deploy both systems and that there is some arbitrage mechanism that determines whether to follow a model-free or a model-based policy at any moment.



A common way of thinking about planning is that it involves a process of searching through a "tree" of possible states that could be encountered. For example, if you are in the kitchen and you are hungry, you might be able to imagine turning left to look for food in the fridge, or right and look in the oven. If you look in the fridge, you might imagine looking on the bottom shelf and recall that there is nothing there, or on the top shelf where there is a tasty apple[96]. Of note, you don't actually need to move a muscle do this – you can easily formulate a plan to go and raid the fridge in search of a snack whilst you are sitting on the sofa watching TV.

Tree search methods, such as monte-carlo tree search (MCTS), involve several computational steps that specify how to choose which states to search (it might be, for example, that you left a tasty pie in the oven). One common method, known as UCB or upper confidence bound, directs the search towards those states whose value is less known. Another problem is how to "back up" the reward once imagined. You need to learn – having imagined obtaining the apple – that the fridge has high value, as does the kitchen itself. Tree search algorithms typically "back up" discounted rewards from each state to its predecessors, in a fashion similar to model-free RL, but occurring in a more explicit fashion.

[96] This is a bad example because apples lose their taste in the fridge.



Task involves making two decision steps to obtain reward, with probabilistic transitions

Shows that humans use a mixture of model-based and model-free policies during a two-step bandit task

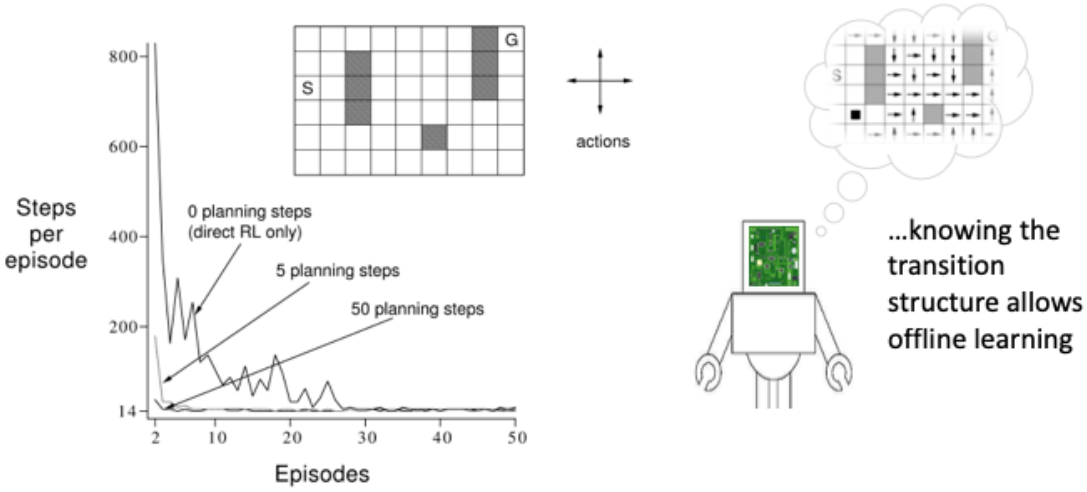
As noted above, model-free and model-based policies have complementary costs and benefits, and it's likely that humans and perhaps other animals use a mixture of the two. Daw and colleagues[97] developed a task that tests for this explicitly, by asking participants to perform a sequential decision or "two-step" bandit task, in which a first-level choice yields no direct rewards but offers the opportunity to transition to one of two distinct states with time-varying payout magnitude. Critically, the authors introduced a manipulation whereby choices at stage 1 occasionally (30% of trials) transitioned to the unchosen level 1 state. Model-free and model-based systems make different predictions about how participant should respond after having made such a rare transition (e.g. choose B, move to C) and been rewarded at level 2 (e.g. for C1). The model free system does not know about the state transition matrix, and thus, will back up the reward to the relevant actions (choose B, choose C1). It thus predicts that on the subsequent trial, all other things being equal, a choice of B is more likely at the start state. However, the model-based system knows that the reward at C1 is more likely to be incurred once again by a choice to C, and thus predicts that this will be more likely on the next trial. As can be seen from the behavioural data, humans acted as if their choices were guided by a mixture of model-based and model-free policies.

There has been a great deal of subsequent research using this task, which cannot be summarised here, but broadly, there is evidence that manipulations that impair prefrontal function, including TMS and stress-testing[98], shift the balance from model-based towards

[97] Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-Based Influences on Humans' Choices and Striatal Prediction Errors. Neuron. 2011;69(6):1204–1215. doi: 10.1016/j.neuron.2011.02.027

[98] Working-memory capacity protects model-based learning from stress
A. Ross Otto, Candace M. Raio, Alice Chiang, Elizabeth A. Phelps, Nathaniel D. Daw
Proc Natl Acad Sci U S A. 2013 Dec 24; 110(52): 20941–20946; The Curse of Planning: Dissecting multiple reinforcement learning systems by taxing the central executive
A. Ross Otto, Samuel J. Gershman, Arthur B. Markman, Nathaniel D. Daw Psychol Sci. 2013 May; 24(5): 10.1177/0956797612463080.

model-free behaviour, consistent with the role of PFC in model-based planning. However, the relative contributions of different brain regions involved in reward-guided decision-making to computing value signals predicted by the two approaches remains somewhat unclear[99].

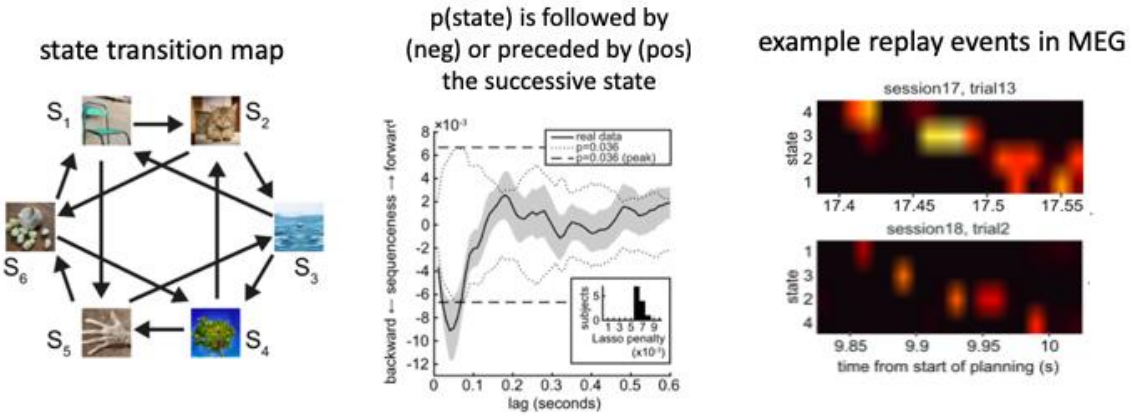


DYNA uses offline replay to accelerate learning, e.g. in a grid world

Sutton 1999

As an aside, it's worth pointing out that there are multiple mechanisms by which imagining possible state transitions might facilitate learning and decision-making. For example, it's possible that the model-based and the model-free system interact. One theory that we have already encountered is that the model-based system might be used to train the model-free system. This is formalised in a machine learning method known as DYNA, in which the agent learns the state transition matrix explicitly but doesn't plan – rather it uses imagined experience to train the model-free system via TD learning or a similar model-free update rule. DYNA greatly accelerated learning on tasks in which there are many states or rewards are sparse, because it allows the agent to learn rapidly from reimaging past experiences.

[99] The ubiquity of model-based reinforcement learning Bradley B Doll, Dylan A Simon, Nathaniel D Daw Curr Opin Neurobiol. 2012 Dec; 22(6): 1075–1081.

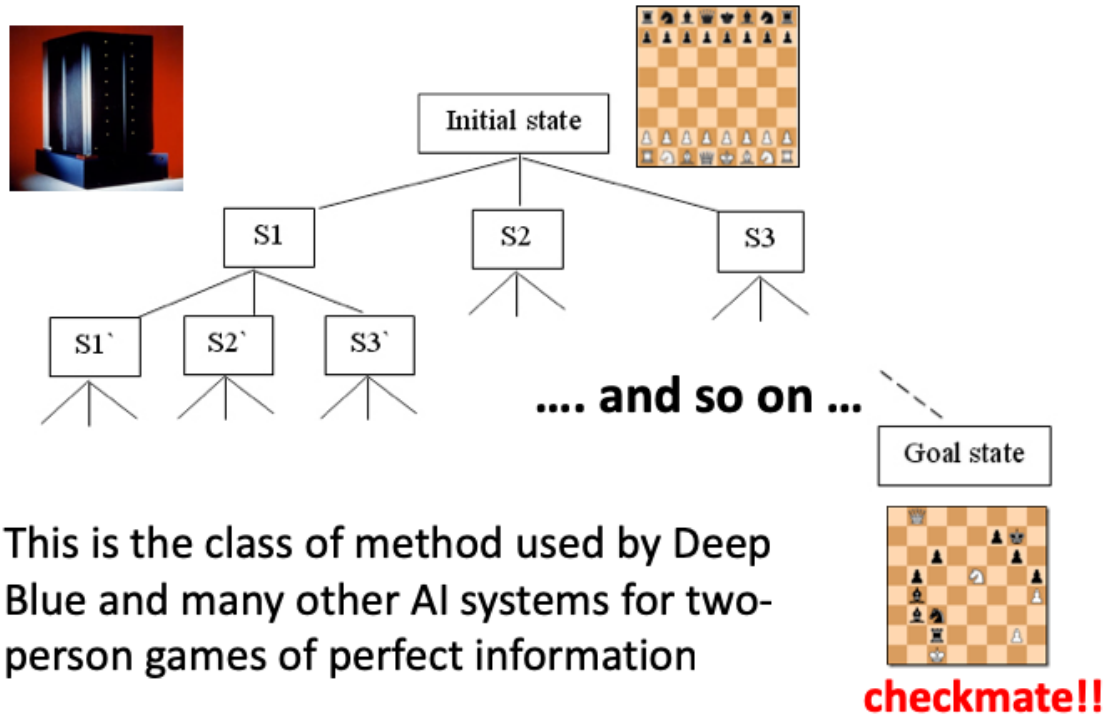


Human replay-like events can be recorded from whole-brain MEG signals

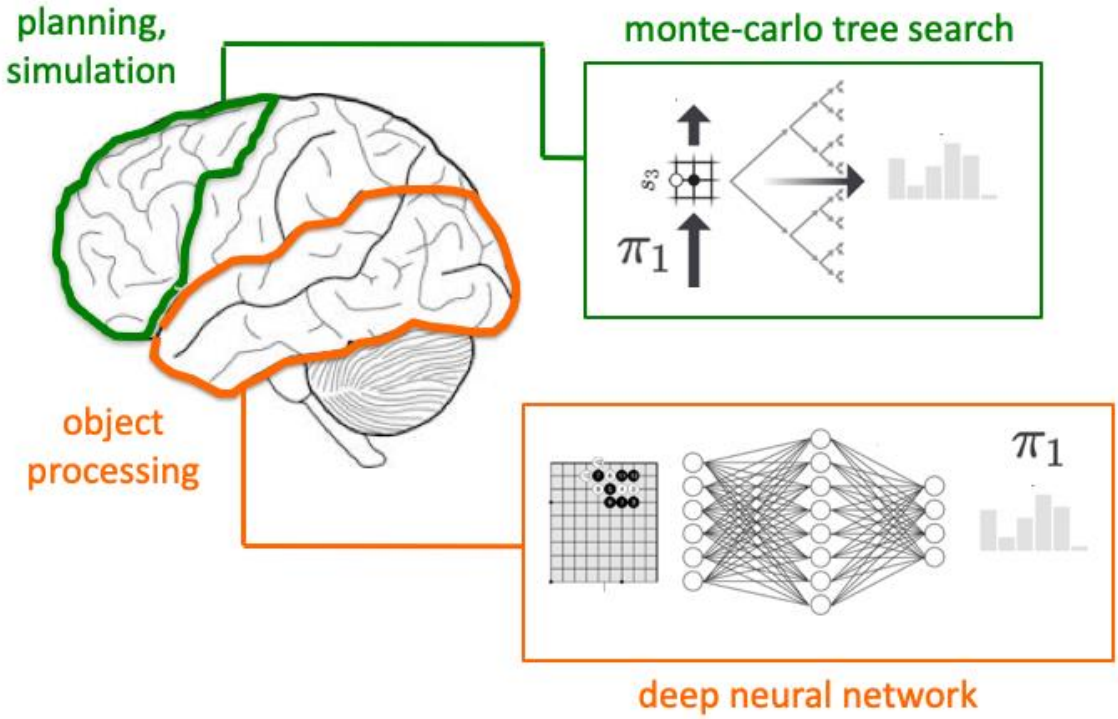
Kurth-Nelson 2016

In lecture 6, we have already discussed the notion that hippocampal replay might involve exploration of a model of the world (e.g. replaying memories to consolidate them from hippocampus to the cortex, either during sleep or quiet resting, or at key points during a task). However, it's also tempting to think of replay as a potential neural marker of a forward or backwards search through a tree of possibilities. This paper from Kurth-Nelson and colleagues[100] suggests that even in humans, it may be possible to identify neural markers of planning that resemble replay (or "preplay"). Participants planned routes through a series of objects which were associated with variable reward, attempting to formulate a plan that maximised their reward. The authors used MEG in concert with multivariate decoded to identify neural signals associated with each of the states, and then decoded these states during planning time, when participants were deliberating about which object route to take. The authors found evidence for backwards replay (with some examples of forwards replay), i.e. neural signals for a state that would subsequently be chosen was more likely to be followed by those for a state that preceded it.

[100] Fast Sequences of Non-spatial State Representations in Humans Zeb Kurth-Nelson, Marcos Economides, Raymond J. Dolan, Peter Dayan Neuron. 2016 Jul 6; 91(1): 194–204.



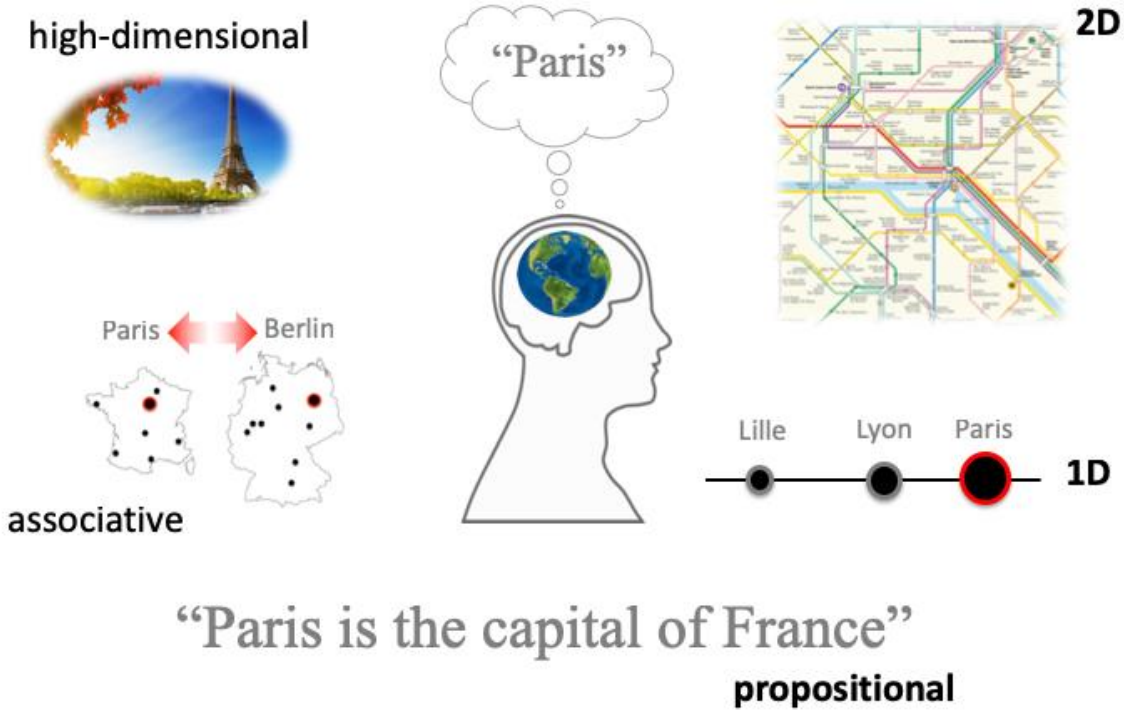
In AI research, tree search has a long history, and we have already encountered search-based algorithms that led to strong performance in complex domains, such as chess.



However, as we discussed, some other problems, such as Go, until recently remained elusive because of the very high branching factor, i.e. the rate at which the tree grew in simulation

(because, for example, of the number of possible positions where a piece could be placed). More recent approaches combine deep reinforcement learning and tree search[101], using a convolutional neural network to learn the value of different board states, and to use this knowledge search more efficiently through the tree of possible moves. As we have seen previously, it is often through combining machine learning approaches to resemble the "modular" brains that have evolved in biological systems that the strongest performance is elicited.

8.3. Cognitive maps and the hippocampus



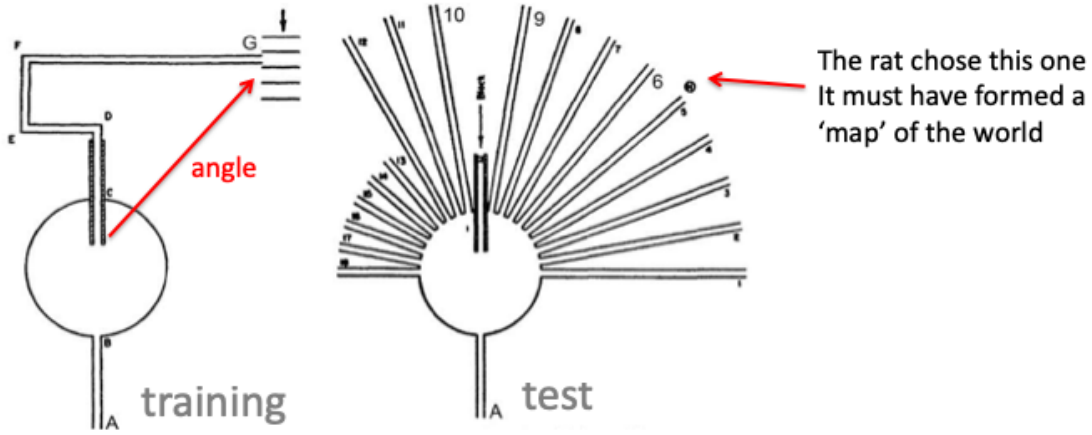
At the start of this course, we discussed how one of the great challenges of building AI systems that display human-like intelligence is the tremendous richness of human knowledge – humans know stuff. What we mean by this is that humans have a rich model of the world, that they can use for mental simulation and planning. However, as we have seen, planning is computationally costly. How can we encode world knowledge in a way that is useful for planning and mental simulation, that avoids the potentially prohibitive costs of imagining every possible outcome?

One argument is that the human mind is special because it has evolved to represent information about the world in a variety of different, but useful formats – and to translate effectively between them when making inferences. For example, think about your knowledge of Paris. You may have been there – in which case you may retain vivid episodic memories of your trip, or a high-dimensional representation of Paris. However, Paris may be represented in your brain in other ways. You. Might associate Paris with Berlin, or with eating snails, or with the Notre Dame – i.e. with other stimuli that you might have co-experienced in various ways.

[101] Find papers, information and more here: https://deepmind.com/research/alphago/

You also have various low-dimensional representations of Paris. You might be able to visualise it on a map, or to recall the subway is laid out, which is a 2D or "allocentric" representation. You even have a 1D representation of Paris – for example, you know that if all the cities in France were ordered according to their size, i.e. on a line, then Paris would be at one end, because it is the largest.

Another way of framing this contention is that we have evolved to represent the world as a series of maps – structured representations that encode information in a useful way. This might allow planning to proceed not just over individual states (such as moves in a game of Go) but also by taking into account the structure – i.e. the topology and geometry – of states in the world[102].

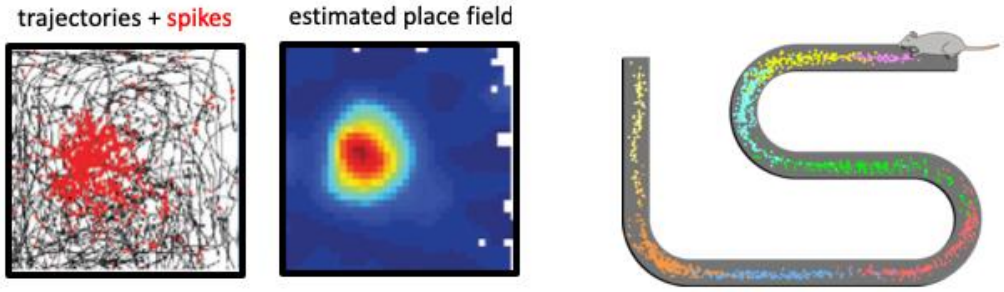


Rodent behavior suggests that they have topological or geometric representations of allocentric space

Tolman 1946; see Behrens 2018

We've known for a long time that even rodents behave as if they form a "map-like" representation of the world, i.e. that they understand the geometry of space in a way that cannot be explained by model-free RL alone. For example, Tolman trained rats to reach a reward by exiting a circular environment and following a passageway with several twists and turns to a goal location. At test, the circular environment was changed to offer multiple exit routes, and the rats on the very first trial chose the arm that would have led directly to the goal taught during the training phase. In other words, the animals can make "zero-shot" inferences about the world given an understanding that space is organised in two dimensions.

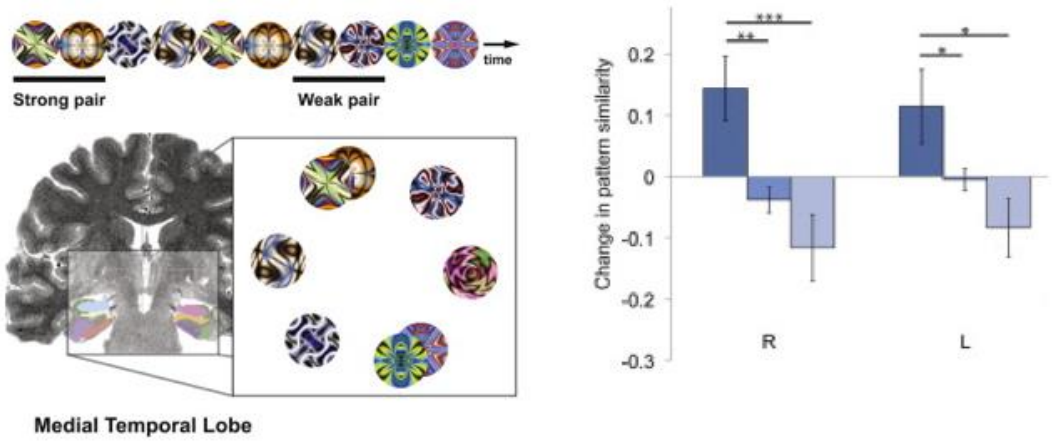
[102] What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. Behrens TEJ, Muller TH, Whittington JCR, Mark S, Baram AB, Stachenfeld KL, Kurth-Nelson Z. Neuron. 2018 Oct 24;100(2):490-509. doi:



Place cells are sparse representations, coding for a unique locations in space as a rodent explores an open arena or runs on a track

O'Keefe & Nadel 1976

We have also known for a long time that one-shot inferences about a goal location are disrupted after hippocampal damage[103]. The hippocampus is an excellent candidate to encode an allocentric map of space, with states encoded in place cells, which fire at a specific location in a testing environment.

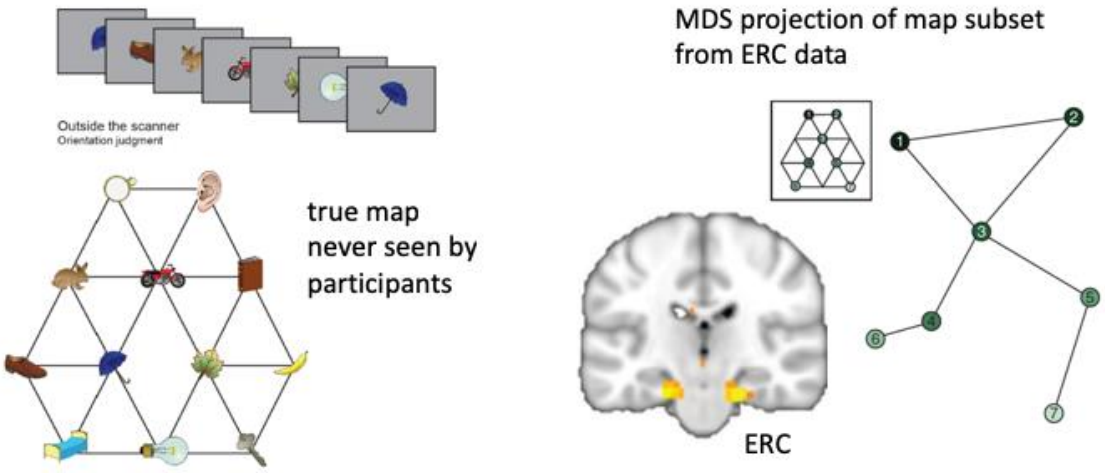


Hippocampal representations become more similar for associated pairs during statistical learning

Shapiro et al 2012; see also Miyashita 1993

[103] http://www.scholarpedia.org/article/Morris_water_maze

One of the requirements of model-based control is that the transition matrix is learned during encounters with the world. There is evidence from a large body of work focussed on episodic encoding into long-term memory that the hippocampus learns associations between words and objects, such as faces and places[104]. However, statistical learning of the transition matrix in the human hippocampus is perhaps most clearly illustrated by this study from Shapiro and colleagues, in which sequences of abstract images (fractals) were presented, with some transitions frequent and others rare. Multivoxel patterns in the hippocampus became more similar for fractals that frequently occurred in succession, as if this structure were encoding the temporal association between adjacent states. Note that similar effects have been reported in the neocortex, but after more prolonged training[105], perhaps indicative of a consolidation process via CLS or a related mechanism.



During statistical learning, map-like representations form in the MTL (here, in ERC)

Garvert et al 2017

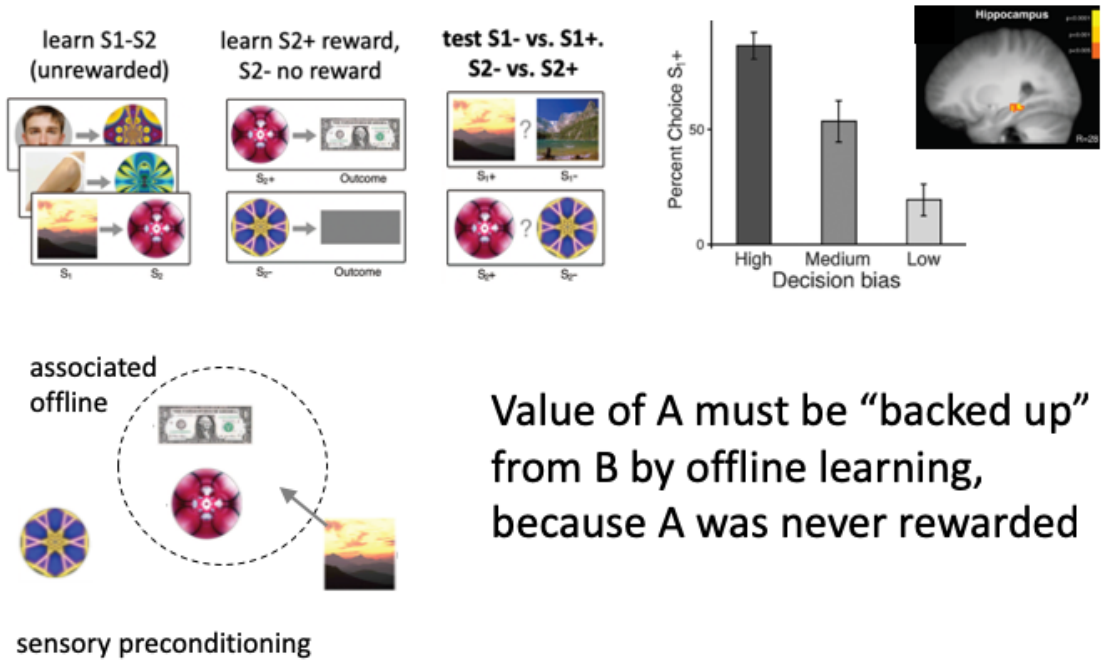
More recent studies have shown that map-like representations exist in other MTL structures, such as the medial entorhinal cortex. In this statistical learning paradigm from Garvert et al, participants first viewed a sequence of objects with transitions determined by a 2D map with hexagonal structure[106]. They were then shown a subset of transitions (object-object pairs) in random order, and repetition suppression (RS) was measured as an index of association. By measuring RS and projecting the data back into 2 dimensions (using MDS) the authors could recapitulate the structure of the map in the medial ERC.

---

[104] Neocortical connectivity during episodic memory formation. Summerfield C, Greene M, Wager T, Egner T, Hirsch J, Mangels J. PLoS Biol. 2006 May;4(5):e128. Epub 2006 Apr 18.

[105] Neural organization for the long-term memory of paired associates. Sakai K, Miyashita Y. Nature. 1991 Nov 14;354(6349):152-5.

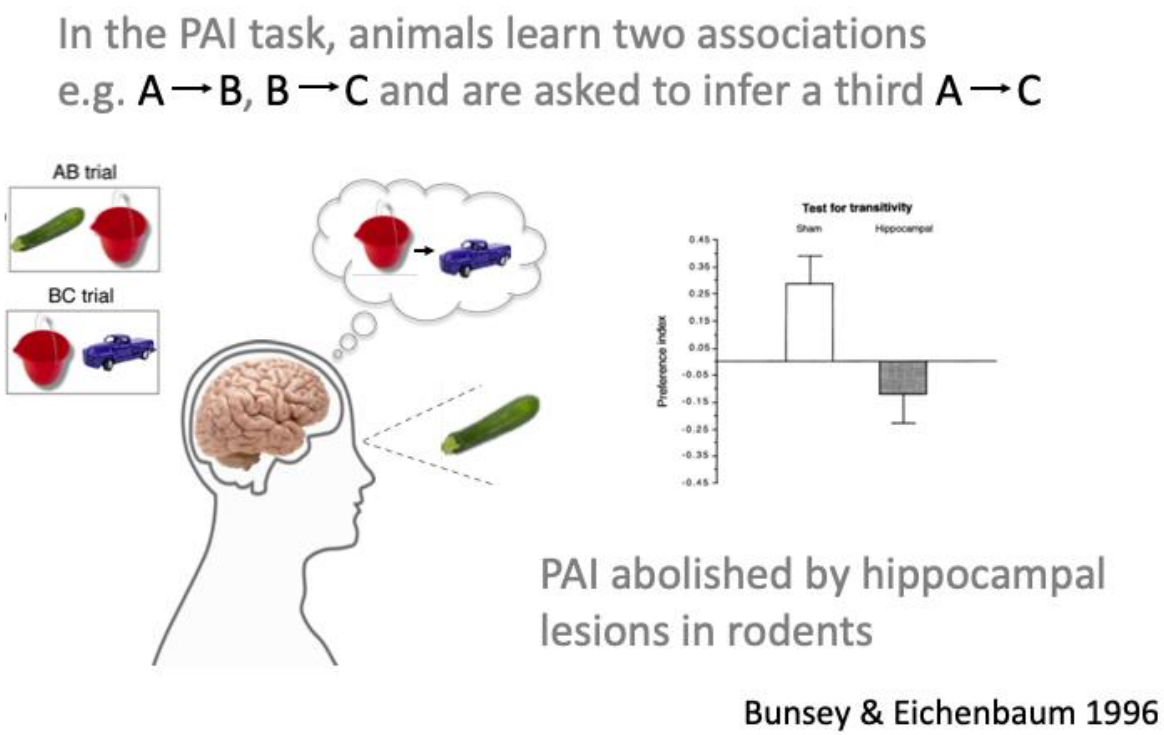
[106] A map of abstract relational knowledge in the human hippocampal-entorhinal cortex. Garvert MM, Dolan RJ, Behrens TE. Elife. 2017 Apr 27;6.



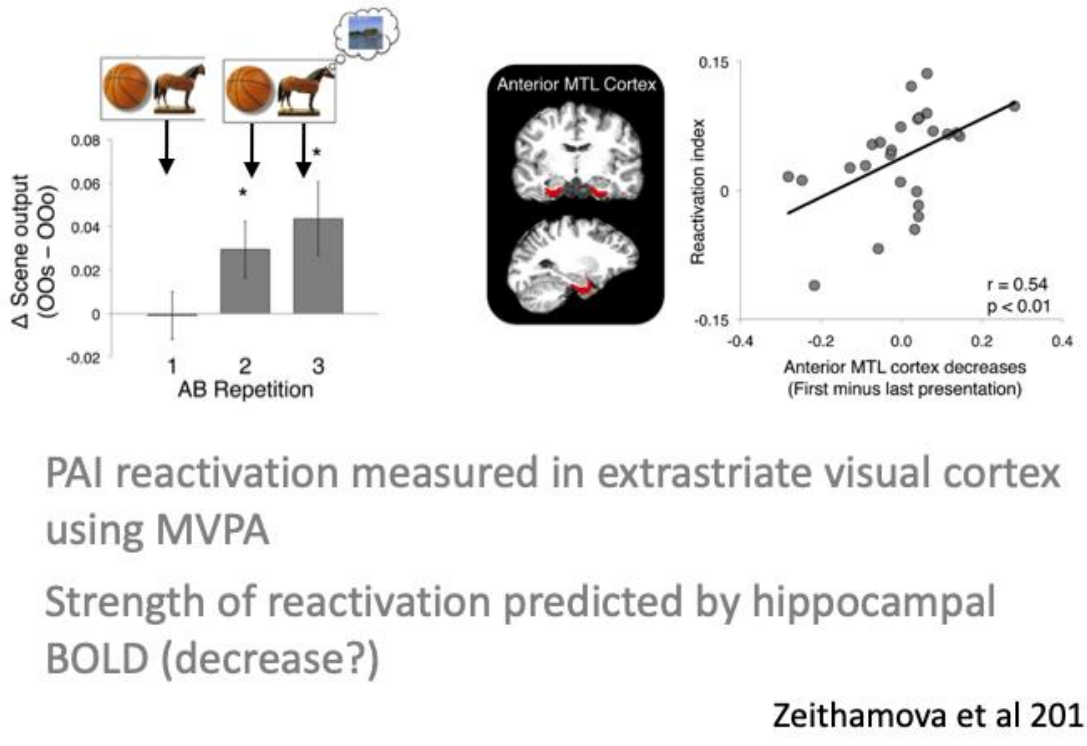
Wimmer et al 2013

Other studies have shown how the hippocampus can be used for inference about rewards using a model of the world. For example, in sensory preconditioning, two stimuli S1 and S2 are first associated without any reward. Subsequently, S2 is paired with a positive or negative outcome. At test, participants are asked to choose between S1 stimuli whose S2 partner was rewarded or not. In this study by Wimmer and colleagues[107], BOLD signals in the hippocampus were stronger for those S1 stimuli associated with rewarded than unrewarded S2 stimuli, at least in those participants that showed the strongest behavioural bias towards S1+ stimuli.

[107] Preference by association: how memory mechanisms in the hippocampus bias decisions. Wimmer GE, Shohamy D. Science. 2012 Oct 12;338(6104):270-3.



Another paradigm that reveals the role of the hippocampus in model-based inference over states is the paired associate inference (PAI) paradigm, in which animals learn two associations (A-B, B-C) and are tested on their knowledge of A-C. PAI is abolished by hippocampal lesions in rodents[108].



[108] Conservation of hippocampal memory function in rats and humans. Bunsey M, Eichenbaum H. Nature. 1996 Jan 18;379(6562):255-7.

8.4. Hierarchical planning

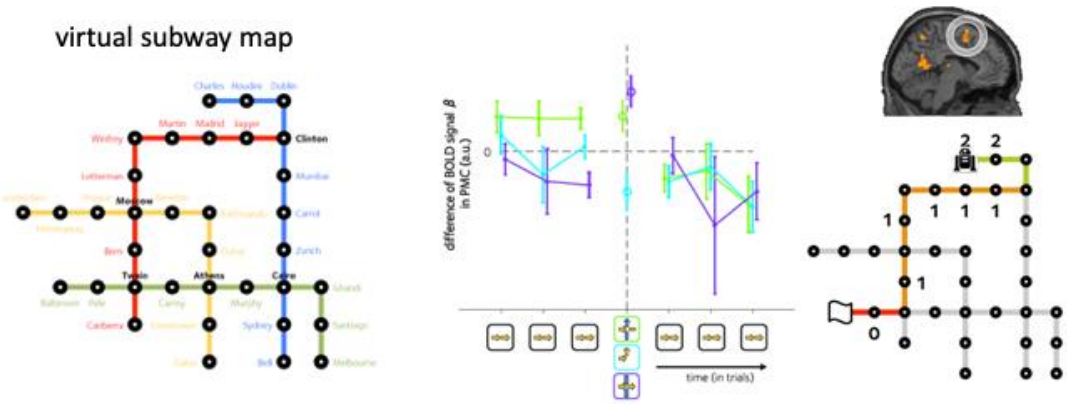
Using a related task in the scanner, Zeithamova et al[109] showed that hippocampal BOLD predicts the degree of reactivation as demonstrated by mutivoxel pattern association between items A and C during the PAI task. However, there are some idiosyncrasies to this study (why does hippocampal BOLD *decrease* predict reactivation index?).

During model-based planning, we don't form policies over individual states, but over chunks or clusters

e.g. "take the piccadilly line to King's Cross, then change"

These experiments show that for simple paired associate learning, the hippocampus seems to encode transitions and allow novel inferences. However, as we have seen, the computational cost of inference grows with the size of the map, i.e. the number of states. Thus, for effective planning, agents need to learn a model of the world that is structured in an efficient fashion, i.e. with representations over multiple spatial and temporal scales. For example, when planning a journey on the London Underground, you don't need to consider. Every single station *en route* to your journey – it's often sufficient to know how which line to take and where to change. In other words, your journey is planned not only over states (e.g. King's Cross) but also over contexts (e.g. the Piccadilly Line). In other words, planning is *hierarchical*.

[109] Hippocampal and ventral medial prefrontal activation during retrieval-mediated learning supports novel inference. Zeithamova D, Dominick AL, Preston AR. Neuron. 2012 Jul 12;75(1):168-79.

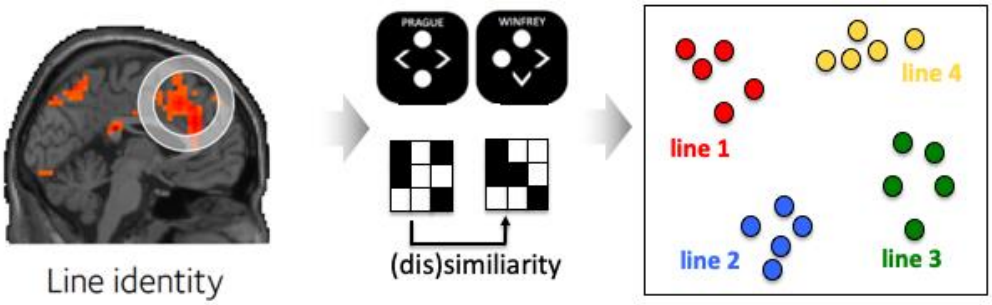


Posterior dACC responds to bottleneck states, and encode distance to goal in number of lines, not states, with an abrupt drop in BOLD at the bottleneck

Balaguer et al 2016

To better understand hierarchical planning, Balaguer and colleagues[110] asked humans to learn to make journeys (from a start state to a goal state) in a virtual subway network. Participants were not shown the map when navigating but had to select buttons to move between stations arranged in a subway-like graph. An dACC region responded with higher signal at the bottleneck states (i.e. the intersections between lines, where a switch was possible). However, this region also encoded the cost of further planning (i.e. the distance to goal) not only in units of states (i.e. the number of stations) but only in units of contexts (i.e. number of lines). In other words, this region may be helpful in using temporal abstraction to form plans.

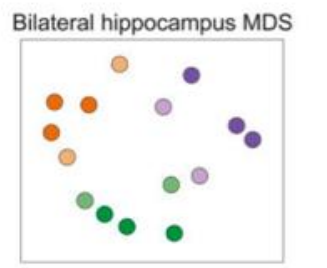
[110] Neural Mechanisms of Hierarchical Planning in a Virtual Subway Network. Balaguer J, Spiers H, Hassabis D, Summerfield C. Neuron. 2016 May 18;90(4):893-903.



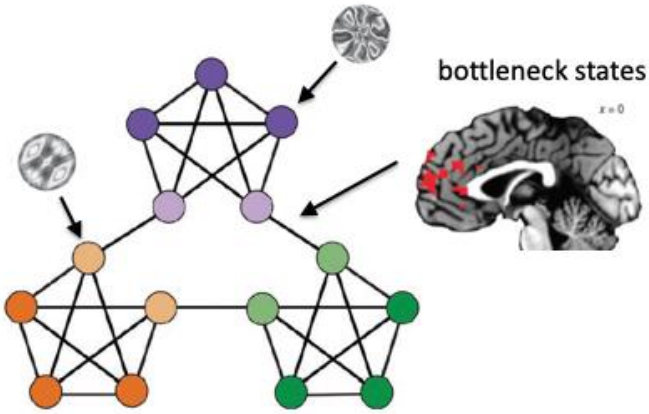
Context (e.g. line identity) can be decoded from the dACC, even thought this was not shown to participants

Balaguer et al 2016

Further evidence for this view comes from the finding that using RSA, the authors were able to decode the line (rather than the station) being currently occupied from the dACC, even though this information was not shown to participants. In other words, the medial PFC may encode a representation of the context over which a plan is being formed, to allow efficient routes to be charted through environments composed of multiple states.



Similar to Garvert 2017

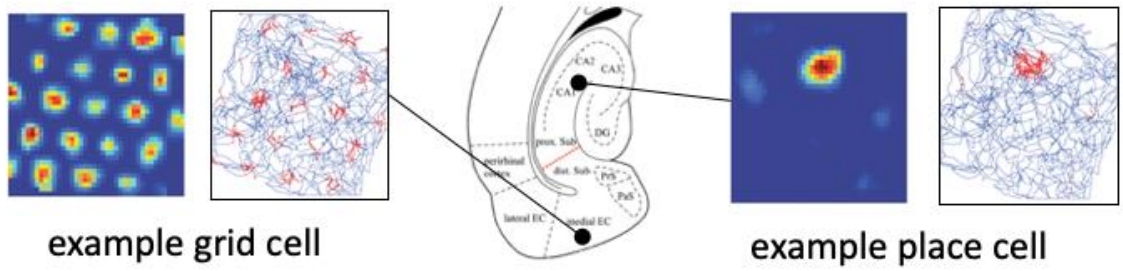


state space with local community structure

Schapiro 2013, 2016

In a related statistical learning study by Schapiro et al[111], participants viewed sequences of fractals that were arranged to have a local community structure, forming "clusters" that were linked by unique bottleneck states. Much like in the Garvert study, the authors could decode a representation of the environment from BOLD signals in the MTL (here, the hippocampus, rather than the ERC), and they similarly found that the dACC responded with higher BOLD signals at the bottleneck states, where 'higher-order' transitions between clusters occurred.

8.5. Grid cells and abstract conceptual knowledge

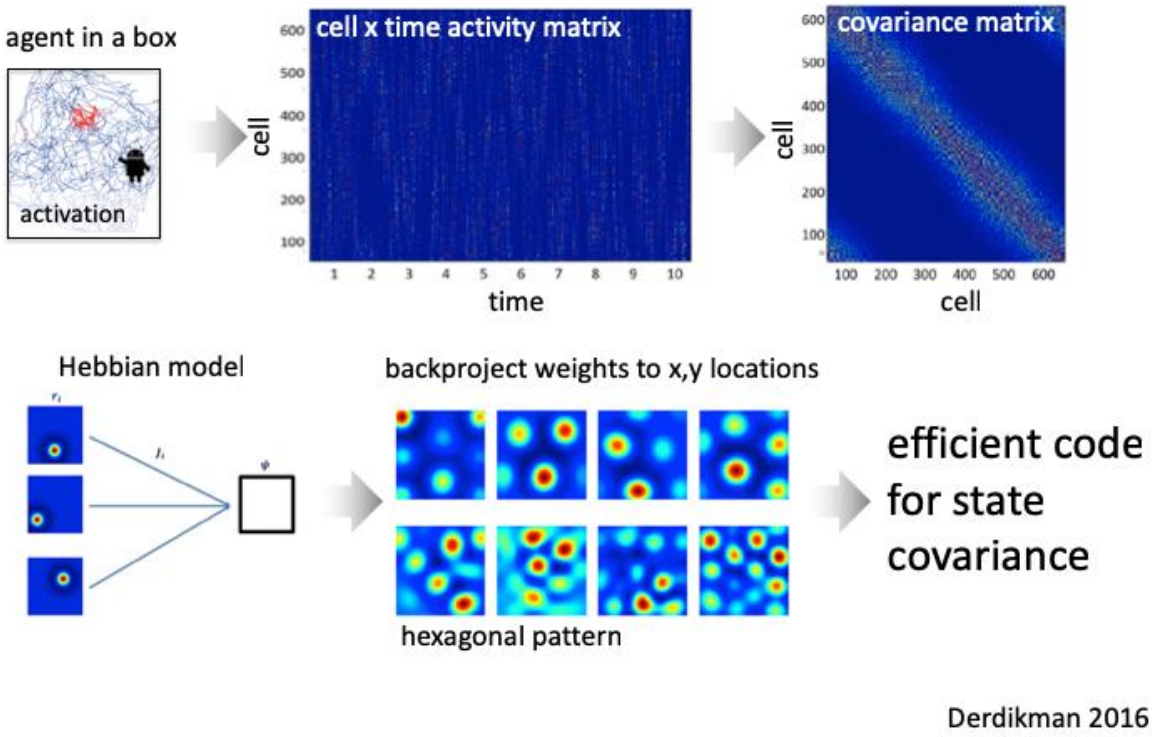


Grid cells are observed in the medial entorhinal cortex of rodents

Moser; O'Keefe; figure from Grossberg, 2013

Finally, let's consider one of the most recent theories of how abstract representations are encoded in the brains of mammals. In addition to place cells (typically observed in the hippocampal CA1/CA3 region), in rodent ERC a different type of cell, known as "grid cells" exhibits a hexagonal lattice-like place field over different scales and phases, firing at regular intervals as the animal explores a testing box. The 2017 Nobel Prize for Medicine was shared between the researchers who discovered place and grid cells.

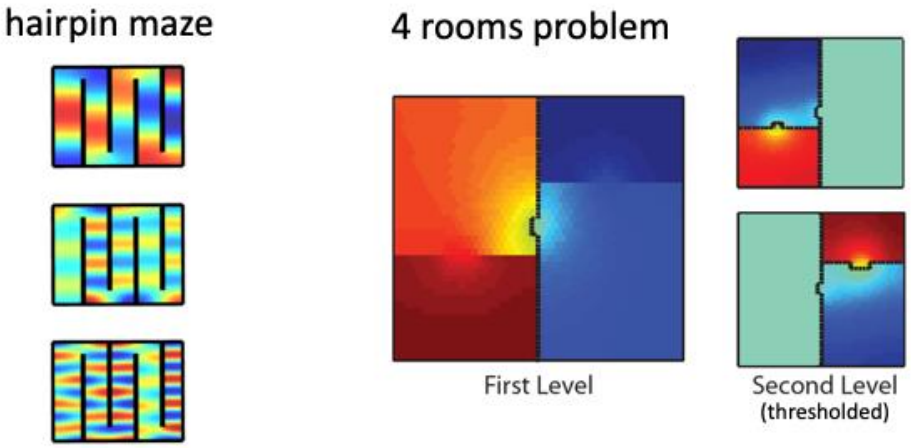
[111] Neural representations of events arise from temporal community structure. Schapiro AC, Rogers TT, Cordova NI, Turk-Browne NB, Botvinick MM. Nat Neurosci. 2013 Apr;16(4):486-92; Statistical learning of temporal community structure in the hippocampus. Schapiro AC, Turk-Browne NB, Norman KA, Botvinick MM.Hippocampus. 2016 Jan;26(1):3-8.



Computationally, one can understand grid cells as encoding the first eigenvector (or principal component) of place cell activity. To illustrate, in this study from the Derdikman lab[112], the authors simulated a rat moving through a box with randomly scatter Gaussian place fields activated in sequence during movement trajectories. If you compute the covariance between each place cell activity and every other, take the first PC, and project back into 2D space, you observe a fourfold lattice; if you additionally impose a constraint that the principal components cannot take negative sign (non-negative PCA), similar to cells which cannot have negative firing rates, then you can recover the same hexagonal lattice that is characteristic of grid cells. Of course, the brain doesn't have immediate access to the full covariance matrix, but we know from previous lectures that Hebbian learning implements an online eigenvalue decomposition according to Oja's rule. The authors simulated this Hebbian learning using a simple shallow network and showed how grid cells could be learned from place cell activation.

If grid cells encode the covariance among place cells, this is important for the learning of abstractions, because it suggest that they form a *relational code*, i.e. encode the pattern of activation among place cells, divorced from the particularities of sensory input[113].

[112] Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. Dordek Y, Soudry D, Meir R, Derdikman D. Elife. 2016 Mar 8;5:e10094.
[113] Navigating cognition: Spatial codes for human thinking. Bellmund JLS, Gärdenfors P, Moser EI, Doeller CF. Science. 2018 Nov 9;362(6415).



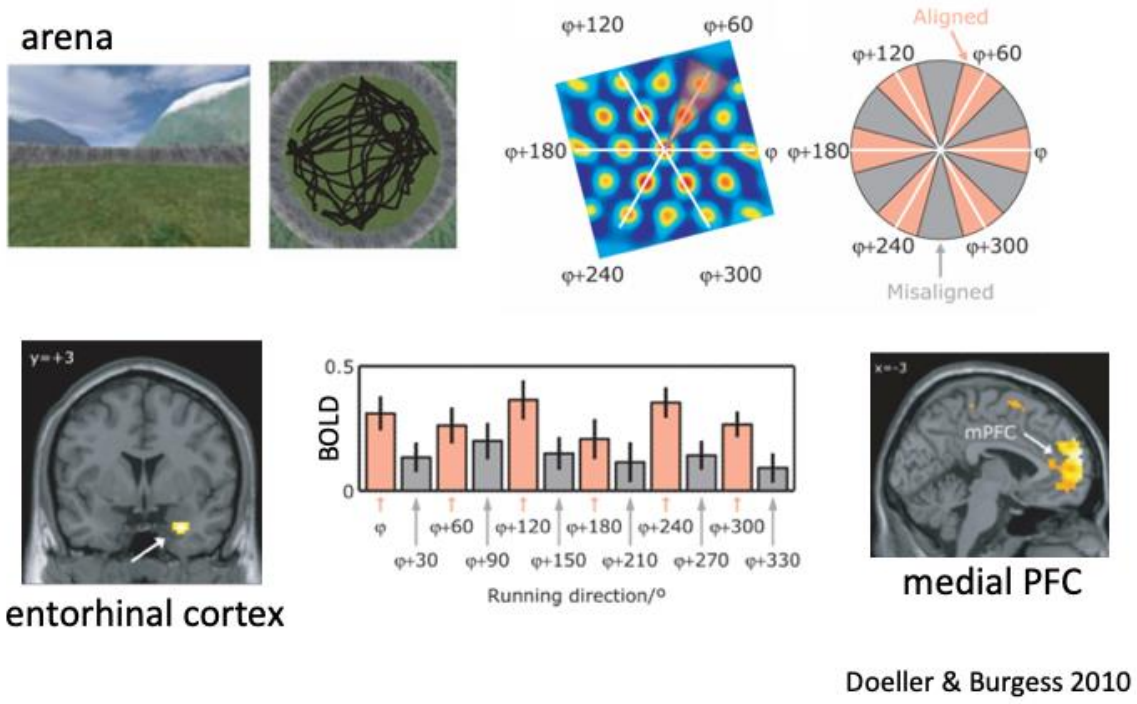
principal components of successor representation chunk the map hierarchically

Stachenfeld 2014

How can we link this notion of a relational, grid cell code for navigation to the need for efficient codes for hierarchical planning discussed earlier? In this theoretical paper by Stachenfeld et al[114], they show how the grid code (because it learns the principal factors of variation in a trajectory of place cell activity) will naturally decompose the representation of space efficiently. For example, moving beyond the open arena considered in the Derdikman paper, if one considers the 4-rooms environment that we started with, a related analysis shows that the principal components (here, of a construct known as the successor representation[115], related to the place code) decompose the environment into a sensible set of representations over multiple scales – the first component identifies one half of the environment relative to the other, and subsequent components index the individual rooms in each half of the environment. In other words, unsupervised learning on place-like representations might allow an agent to learn a hierarchical representation of a given environment, facilitating planning over multiple scales simultaneously (e.g. over rooms and loci within a room). A similar analysis applied to a hairpin maze identified corridors and turnings, followed by loci within a corridor, etc.

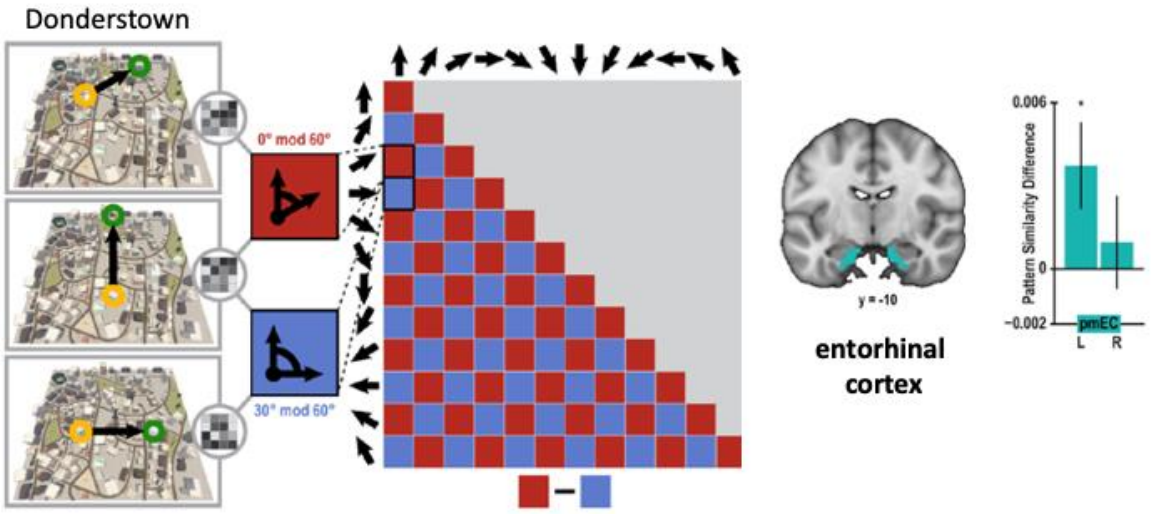
[114] The hippocampus as a predictive map. Stachenfeld KL, Botvinick MM, Gershman SJ. Nat Neurosci. 2017 Nov;20(11):1643-1653.

[115] The Successor Representation: Its Computational Logic and Neural Substrates. Gershman SJ. J Neurosci. 2018 Aug 15;38(33):7193-7200.



Using a clever fMRI approach, Doeller and Burgess[116] showed that it is even possible to identify a signature of grid cell activity in humans. Participants navigated through an open arena, and the authors reasoned that if grid cells exhibit habituation – a suppression of firing after they have been active – then successive trajectories that differ by in-phase multiples of 60° should elicit reduced activity relative to those that differ by out-of-phase multiples of 60°. They found this pattern not only in the BOLD signal in ERC, but also in other regions potentially important for memory and model-based inference, such as the vmPFC.

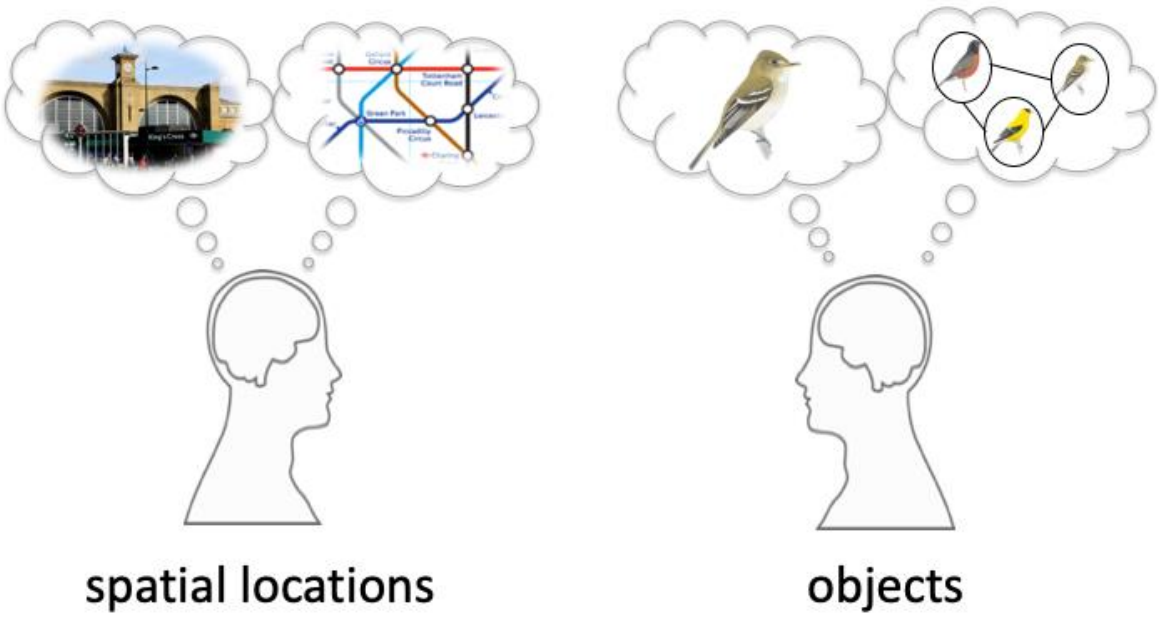
[116] Evidence for grid cells in a human memory network. Doeller CF, Barry C, Burgess N. Nature. 2010 Feb 4;463(7281):657-61.



Six-fold symmetry in human BOLD signals

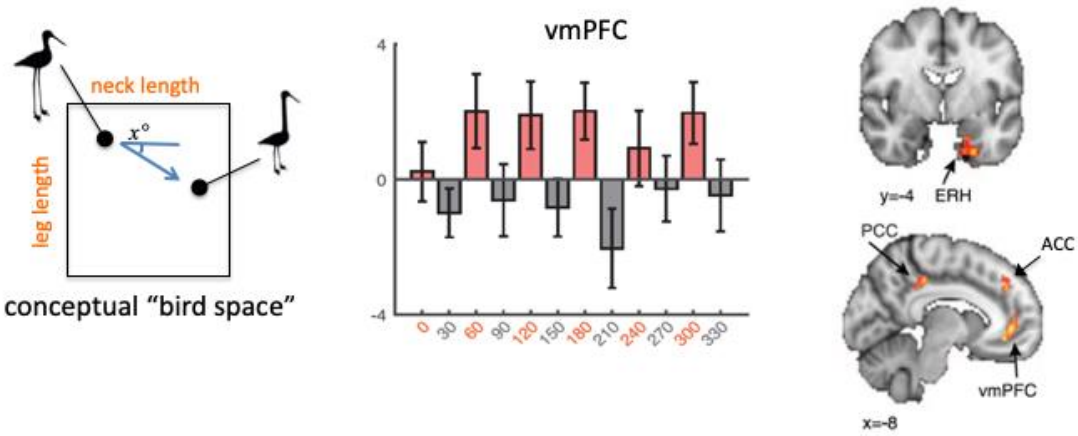
Bellmund et al 2016

ERC grid cells may be important for mental simulation: when participants were asked to imagine making trajectories through a virtual environment ("Donderstown") to one of 12 locations arranged hexagonally from the start point, the ERC grid code in BOLD was observed at the time of planning, before any movement took place[117].



[117] Grid-cell representations in mental simulation. Bellmund JL, Deuker L, Navarro Schröder T, Doeller CF. Elife. 2016 Aug 30;5.

All of this evidence comes from spatial cognition. It might be that grid-like codes encode relational information for space, but not for the potentially more complex relational structures that link objects, individuals, and abstract concepts. However, as we have seen, we can think of objects as organised in a "map" as well – for example, the taxonomic map that organises animals according to their phylum, class or species. Recently, evidence has emerged for sixfold symmetry in the coding of time and directional eye movements[118], two nonspatial domains.



Grid-like representations may encode abstract conceptual spaces

Continescu et al 2016

Perhaps the best evidence for this comes from a recent paper[119] in which participants viewed (and imagined) trajectories through a 2D "bird space" defined by leg and neck length. The authors replicated the findings of Doeller and Burgess in this nonspatial domain, suggesting that grid codes may permit an efficient representation of the covariance more abstract spaces, including those that are learned de novo and do not relate to spatial navigation.

[118] Hexadirectional coding of visual space in human entorhinal cortex. Nau M, Navarro Schröder T, Bellmund JLS, Doeller CF. Nat Neurosci. 2018 Feb;21(2):188-190; Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit. Aronov D, Nevers R, Tank DW. Nature. 2017 Mar 29;543(7647):719-722.
[119] Organizing conceptual knowledge in humans with a gridlike code. Constantinescu AO, O'Reilly JX, Behrens TEJ. Science. 2016 Jun 17;352(6292):1464-1468.